Vacuum Ultraviolet Laser Induced Breakdown Spectroscopy (VUV-LIBS) For Pharmaceutical Analysis

Submitted by:

Muhammad B. Alli

B. Sc.

A thesis submitted for the Award of:

DOCTOR OF PHILOSOPHY

Presented to:

The School of Physical Sciences

Dublin City University

Supervisors

Dr. Patrick Hayden

Prof. John T. Costello

September 2021

Declaration

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of Doctor of Philosophy is entirely my own work, and that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge breach any law of copyright, and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

Signed: ________________

(Candidate) ID No.: 10322935

Date: 02/09/2021

For my grandmother Hajiani Salmaa Bee Bee Alli.

Acknowledgements

I would like to thank my supervisors Dr. Patrick Hayden and Prof. John Costello for the opportunity to carry out this research and the years of support and guidance. I would like to thank all the support staff that have helped me over the years, Des Lavelle, Lisa Peyton and Irene Ryan. I would like to thank my fellow postgrads Hu Lu, Syedah Sadaf Zehra and Tejaswi Katravulapally.

I would like to thank my mother and father for all their support. I would like to thank my friends Abdullah Azaz, Kierain Ryan and William Hanks for always being there.

Table of Contents

List of Figures

List of tables

List of Abbreviations

API	Active pharmaceutical ingredient
BI-CCD	Back illuminated charge coupled device
CCD	Charge coupled device
CE	Coronal equilibrium
CNN	Convolutional neural networks
COMP	Competitive learning
CRE	Collisional radiative equilibrium
DCU	Dublin City University
EI	Expected improvement
GCA	Glass capillary array
HPLC	High performance liquid chromatography
LIBS	Laser induced breakdown spectroscopy
LOD	Limit of detection
LTE	Local thermodynamic equilibrium
MOS	Metal oxide semiconductor
Nd:YAG	Neodymium doped yttrium aluminium garnet
PAT	Process analytical technology
PC	Principal component
PCA	Principal component analysis
RBF	Radial basis function
ReLU	Rectified linear unit
SBR	Signal to background ratio
SGDM	Stochastic gradient descent with momentum
SNR	Signal to noise ratio
SOM	Self-organizing maps
STD	Standard deviation
STFT	Short time Fourier transform
SVM	Support vector machines
TISR	Time integrated spatially resolved
TRLIBS	Time resolved laser induced breakdown spectroscopy
TTL	Transistor-transistor logic
VUV	Vacuum ultraviolet
GPU	Graphics processing unit

Abstract

Laser induced breakdown spectroscopy (LIBS) allows quick analysis to determine the elemental composition of the target material. Samples need little\no preparation, removing the risk of contamination or loss of analyte. It is minimally ablative so negligible amounts of the sample is destroyed, while allowing quantitative and qualitative results. Vacuum ultraviolet (VUV)-LIBS, due to the abundance of transitions at shorter wavelengths, offers improvements over LIBS in the visible region, such as achieving lower limits of detection for trace elements and extends LIBS to elements\samples not suitable to visible LIBS. These qualities also make VUV-LIBS attractive for pharmaceutical analysis.

Due to success in the pharmaceutical sector molecules representing the active pharmaceutical ingredients (APIs) have become increasingly complex. These organic compounds reveal spectra densely populated with carbon and oxygen lines in the visible and infrared regions, making it increasingly difficult to identify an inorganic analyte. The VUV region poses a solution as there is much better spacing between spectral lines. VUV-LIBS experiments were carried out on pharmaceutical samples. This work is a proof of principle that VUV-LIBS in conjunction with machine learning can tell pharmaceuticals apart via classification.

This work will attempt to test this principle in two ways. Firstly, by classifying pharmaceuticals that are very different from one another i.e., having different APIs. This first test will gauge the efficacy of separating into different classes analytes that are essentially carbohydrates with distinctly different APIs apart from one another using their VUV emission spectra. Secondly, by classifying two different brands of the same pharmaceutical, i.e., paracetamol. The second test will investigate of the ability of machine learning to abstract and identify the differences in the spectra of two pharmaceuticals with the same API and separate them. This second test presents the application of VUV-LIBS combined with machine learning as a solution for at-line analysis of similar analytes e.g., quality control.

The machine learning techniques explored in this thesis were convolutional neural networks (CNNs), support vector machines, self-organizing maps and competitive learning. The motivation for the application of principal component analysis (PCA) and machine learning is for the classification of analytes, allowing us to distinguish pharmaceuticals from one another based on their spectra. PCA and the machine learning techniques are compared against one another in this thesis. Several innovations were made; this work is the first in LIBS to implement the use of a short-time Fourier transform (STFT) method to generate input images for a CNN for VUV-LIBS spectra. This is also believed to be the first work in LIBS to carry out the development and application of an ellipsoidal classifier based on PCA. The results of this work show that by lowering the pulse energy it is possible to gather more useful spectra over the surface of a sample. Although this yields spectra with poorer signal-to-noise, the samples can still be classified using the machine learning analytics. The results in this thesis indicate that, of all the machine learning techniques evaluated, CNNs have the best classification accuracy combined with the fastest run time. Prudent data augmentation can significantly reduce experimental workloads, without reducing classification rates

1 Introduction

In this chapter the aims and objective of the present work are outlined and discussed in section 1.1. This chapter introduces laser induced breakdown spectroscopy (LIBS), discusses its application in the vacuum ultraviolet (VUV) spectral wavelength region in section 1.2 and the advantages of VUV-LIBS for pharmaceutical analysis are outlined. Methodologies for the acquisition of spectra over the surface of pharmaceutical tablets as well as pitfalls to expect were leveraged from the literature. Pharmaceutical tablets are solids, often formed by compressing a powder which contains the active pharmaceutical ingredient (API) and an inactive filler. The use of fillers provides a delivery mechanism as the size of a tablet consisting of only the therapeutic amount of API may be too small to easily handle as a consumer product. The fillers can also be formulated to present a pleasant taste for tablet consumption, give a controlled release, etc. It is the API that has the intended therapeutic effect e.g., paracetamol for mild to moderate pain relief. Each pharmaceutical tablet will have custom, and often proprietary, properties (such as thermal conductivity, chemical composition of the entire tablet, intended homogeneity of the tablet, colour, hardness etc.) that will be determined by the manufacturing process and the intended distribution and use. Typical the physical properties of these tablets are engineered to ensure that they are sufficiently robust to withstand the mechanical shock associated with manufacturing, shipping and customer handling, and that the required bioavailability and solubility are maintained to achieve the desired physiological effect.

The general characteristics of laser produced plasmas, the bedrock of LIBS, are described in section 1.3. Laser plasma formation is introduced in section 1.3.1, with and the underlying atomic processes in section 1.3.2. In section 1.3.3 thermodynamic equilibrium in a plasma is introduced to connect the concepts of plasma temperature and the population of charge states at those temperatures. These sections are provided to give a glimpse of the complexity that underlines an observed LIBS spectrum, and to show the need for multivariate analysis. A whole spectrum approach to LIBS is described in section 1.5, in particular principal component analysis (PCA) is discussed in section 1.5.1 as this was selected as the sole technique for multivariate analysis in this thesis. The use of machine learning in the literature with LIBS, to replace multivariate analysis, is outlined in section 1.5.2 and will be explored in more detail in chapter 3.

1.1 Aims and objectives of the present work

The aims and objectives of the research in this thesis are as follows:

- To optimize a VUV-LIBS experimental setup to capture spectra of various pharmaceuticals for analysis.
- To investigate the application of individual computational techniques capable of distinguishing between pharmaceuticals based on their spectra.
- To optimize and compare those computational techniques. To optimize and compare those computational techniques. By comparing the classification fidelity and the nuances of their implementation an optimal use case for each technique for pharmaceutical VUV-LIBS can be outlined.

Quality by design is an approach that aims to ensure the quality of medicines by employing a risk-management methodology that is statistical and analytical, in the development and manufacturing of medicines [1]. Process analytical technology (PAT) facilitates this by being an online/at line quality assessment approach [2]. Recently the focus has increased in the pharmaceutical industry for analytics, particularly in PAT. This interest has been noted both domestically in Ireland [3-5] and internationally [6-27], ranging from medical research in cardiovascular medicine and cancer [9-16] to pharmaceutical research [17-27].

The increased research focus on PAT stems from wanting an improvement in manufacturing processes, namely quality control. For a PAT important information to ascertain could be API distribution [28], coating thickness/uniformity [29] and, where the API is a complex molecule, that the correct molecular form of the API has been included. Visible LIBS has already achieved rapid at-line analysis of tablet coating and uniformity [29], however at-line minimally destructive analysis is the criteria considered for the application of this work for PAT for VUV-LIBS. As no sample preparation is required, and the sample is measured with minimal destruction, the product is measured as produced (rather than as prepared for measuring) with the enhanced performance of VUV-LIBS.

It should be noted that this work is not focused on methodologically controlling and characterising the constituents and properties of tablets to observe changes in LIBS spectra. Rather this work is a proof of principle that VUV-LIBS in conjunction with machine learning can tell pharmaceuticals products that are available to a consumer apart via classification. This work will attempt to test this in two ways. Firstly, by classifying pharmaceuticals that are very different from one another i.e., having different APIs. This first test will gauge the efficacy of separating analytes that are essentially carbohydrates with distinctly different APIs apart from one another. Secondly by classifying two different brands of the same pharmaceutical, i.e., paracetamol. The second test will investigate the ability of machine learning to abstract and identify the differences between two pharmaceuticals with the same API and separate them.

While this could appear to be of more use for detecting fraudulent or adulterated products, the same principles would lead to quick reaction times in adjusting any production step to provide the desired product if used as an at-line tool in the manufacturing process. In addition, the sensitivity of VUV-LIBS would allow monitoring at the start of runs to ensure that the is no cross contamination from previous pharmaceuticals manufactured in a previous run on the same production line. Machine learning could be beneficial to PAT and VUV-LIBS due to the speed of analysis. Once models are trained classification with their deployed versions is fast. A portion of the work in this thesis is also to reduce the number of spectra required to train the various models, decreasing the time required for this step. The ability of these techniques to abstract and find differences in an automated fashion, as opposed to a time-consuming manual feature engineering phase, is also a key benefit.

Typically, LIBS is used for quantitative analysis by studying the intensity of spectral lines to determine concentrations. LIBS can also be used for qualitative analysis as, since each element and ion-stage in a sample produces a unique spectral signature, elements present in the target material can be identified. The analysis in this thesis is *neither quantitative nor qualitative* in a traditional LIBS sense. The analysis in this thesis is instead classification analysis, which is distinguishing between analytes based on whole spectra as opposed to lines of interest selected from a spectrum. The main competition, as a technique in PAT, for LIBS is the already established separation method of high-performance liquid chromatography (HPLC). HPLC is chromatography executed at high a column pressure. HPLC is the standard in industrial laboratories as it has a very good recovery of analysed pharmaceuticals put into the process, comparable to LIBS which is usually minimally ablative.

A strong advantage inherent in LIBS is that there is no need for solvents or other material preparation; however, its usefulness at a given point on a production line is dependent on whether atomic or elemental information is sufficient over molecular information. This drives the biggest difference between HPLC and LIBS, HPLC can yield molecular information [30, 31]. A disadvantage of HPLC is that it requires the entirety of an analyte to be dissolved prior to analysis. This means there is no local information that can be gained, only a measure of the entire dissolved sample [32]. LIBS allows sampling of individual points on the sample, allowing the homogeneity or distribution of ingredients to be measured. LIBS, if setup for depth profiling, even allows sampling while boring into the sample with sequential laser shots [33].

X-ray fluorescence (XRF) was reviewed at the outset of this work, as an advantage of XRF is that, unlike LIBS, it is completely non-destructive. A major drawback, which kept it from being selected for further investigation, is that XRF is a surface analysis tool and cannot easily analyse light elements such as carbon and oxygen [34]. This is an issue as carbon and oxygen are the major constituents of pharmaceutical tablets.

Raman spectroscopy was also reviewed. If correctly performed Raman spectroscopy is non-destructive which is an advantage over LIBS. A drawback is that the Raman effect can be very weak leading to long analysis times, and fluorescence by impurities or the pharmaceutical tablets themselves may obscure the Raman spectra [35]. For this reason, Raman spectroscopy was not investigated further in this study.

VUV-LIBS was selected as the analytical technique of choice in this thesis as it has none of the mentioned disadvantages of HPLC, XRF or Raman. The advantages of VUV-LIBS over NIR and VIS LIBS are described in section 1.2.

This work was the first of its kind in the labs at DCU – to use pharmaceutical samples in the VUV-LIBS system and to introduce machine learning techniques. My contributions to this work were extensive and included:

- The acquisition of all experimental spectra shown in this thesis, along with implementing the necessary experimental optimisations needed to do so, detailed in section 2.2.
- Introducing the novel approach to implement a short-time Fourier transform (STFT) method, detailed in section 3.3, to generate input images for a convolutional neural network (CNN) for VUV-LIBS spectra.
- The development and application of an ellipsoidal classifier for PCA, as detailed in section 5.2.2.
- Demonstrated that the surface preservation of analytes can be achieved by reducing laser fluence, detailed in section 5.3.3. Although this yields spectra with poorer signal-to-noise, they can still be classified with machine learning analytics, as shown in the results of sections 5.1.1 and 5.1.2.

1.2 Laser induced breakdown spectroscopy

LIBS presents itself as an attractive method to analyse pharmaceuticals. As an analytical technique, LIBS allows for quick analysis to determine the composition of the target material. Samples need little or no preparation, thereby removing the risk of contamination or loss of analyte. It is normally minimally ablative and so a negligible amount of the sample is destroyed, and it can provide both quantitative and qualitative results. Samples can be in the solid, liquid or gas phases and they can be semiconductors, conductors or insulators [36].

LIBS entails focusing a laser pulse or pulses onto a sample, where the laser pulse energy is absorbed by the sample, thereby generating a plasma. During the period that the pulse is incident on the surface of a sample the resultant plasma ionisation and excitation of the bound atomic electrons are high. It is also the period that the highest plasma temperatures and densities are achieved which, combined with the excitation of free electrons, lead to the observation of a background continuum. The continuum occurs due to bremsstrahlung (free-free) and recombination (free-bound) processes. As the free electrons have a continuum of energies, so too does the radiation emitted. This broadband light contains little useful spectroscopic information.

As the plasma decays, cooling and becoming less dense, the continuum will decay quicker than the spectral lines. It is the spectral information relating to the line emission that is of most interest in LIBS; as the wavelength of this emission is a unique signature of the atoms and ions present in the plasma, and thus the sample from which it was formed. The identification of these emission lines and the quantification of their intensities, from which elements may be classified and potentially quantified, forms the basis of LIBS [37].

VUV-LIBS [38,39] has three key advantages over traditional LIBS in the visible-near infrared (Vis-NIR) spectral region, defined as the wavelength range between 380 nm and 1400 mm [40]. Firstly, it allows access to resonance lines of atomic species, those where the lower level involved in the transition is the ground state, which usually have one of the highest, if not the highest, oscillator strengths. This in turn means that, in principle, lower concentrations of trace elements can be measured – a lower limit of detection (LOD) is possible. Secondly, the VUV region (between 10-200 nm [40]) opens access not just to resonance lines of atoms but also of ions, for which the resonance wavelengths move to shorter wavelengths with increasing ionisation stage. This opens up even more strong lines for LIBS. Finally, laser plasma VUV spectra tend to be less cluttered than their Vis-NIR counterparts and hence provide for the detection of well isolated lines [41]. Vis-NIR spectra exhibit many lines due to supernumerary transitions between excited states of atoms and ions.

As will be discussed in section 1.3.1, plasma consists of a distribution of charge states, depending on the temperature and electron density within the plasma. Therefore, ions will always be produced in laser produced plasma. The benefit of being able to observe emission from these ions is that the transitions observed can, often, be stronger that in atoms. In addition, the ionisation limit, the amount of energy required to free the outer most electron, for ions is higher compared to neutral species. This higher energy allows resonant transitions at shorter wavelengths, suitable for VUV spectroscopy, thus increasing the spectral range over which lines can be observed when compared to a scenario where only neutral species could be observed.

1.2.1 LIBS literature review

Laser induced plasmas were observed shortly after the invention of the ruby laser in 1960 [42]. Debras-Guédon and Liodec published the first use of spectra from laser produced plasmas for spectrochemical analysis of surfaces in 1963 marking the beginning of LIBS [43]. Chemometrics is the use of statistical principles and mathematical models to derive results in analytical chemistry, the first use of chemometrics in LIBS was by Wisburn et al., published in 1994, involving the detection of trace elements in solid environmental samples [44]. The first use of LIBS for space exploration dates to the 1986 work of M. D'Orazio et al., which was conducted to provide a method for the chemical analysis of asteroids and comets [45]. The main instrument in the study were Raman and mass spectrometry with a UV spectrometer was included to observe emission from a laser plasma.

ChemCam, a LIBS based remote analysis system, was incorporated into the Curiosity Mars rover and was capable of interrogating samples 7 m from the rover [46, 47]. The first LIBS spectra were observed on Mars in 2012.

LIBS has been extensively used for the elemental analysis of steel [48-69] and has also been used for the elemental analysis of halogens and metals in a variety of targets [70-74, 28]. LIBS has even been used to analyse the elemental composition of tea [75].

1.2.2 LIBS pharmaceutical analysis literature review

LIBS has demonstrated potential for quantitative analytics with pharmaceuticals in the characterisation of aerosolized drugs [76] and calcium tablets [77], as well as monitoring blending uniformity when LIBS is used in conjunction with inductively coupled plasma optical emission spectrometry [78]. LIBS has been used to determine the concentration of nutrients in a herbal medicine [79] and hyperspectral images of depth profiled tablets can be generated to demonstrate homogeneity in those targets [80]. Pharmaceutical analytes had their concentration measured with LIBS and HPLC, with LIBS showing agreement with the more established industry standard HPLC, demonstrating a high correlation ($R^2 = 0.9866$) [81].

LIBS has demonstrated its capability as a technique to analyse film coatings via depth profiling [28], important quality as most pharmaceutical tablets have a coating on them. The coatings can provide, for example, better taste, make them easier to swallow and give a more palatable appearance. These coatings can also be used to host additional drugs and provide protection from air, moisture and light. The coatings can also be designed to release drugs at a certain time or rate. Also, inside pharmaceutical tablets, there is an issue with the homogeneity of active ingredients within the tablet, and their dispersal. Pharmaceutically engineered biomolecules are becoming increasingly complex, and a large concern is the reduction in solubility and bioavailability, exasperated by the larger molecule size and increased inhomogeneity.

An important note is that the coatings on pharmaceutical tablets need to have the correct thickness, uniformity and chemical composition to achieve the desired effect. They must be able to generate the precise conditions to activate/release an active pharmaceutical ingredient at the right time to ensure its bioavailability where it counts. Profiling the thickness and composition of these layers has been an active area of pharmaceutical LIBS research [82-89], a recent study in this vein of research has shown LIBS' viability in the detection of toxicity in oral antidiabetic drugs [90].

Important physical parameters related to this LIBS study include pulse energy, pulse to pulse variation, energy density, thermal conductivity of tablet, chemical composition of the sample, homogeneity of sample, hardness, size of melt zone and curvature of the sample. The experimental parameters relating to the laser pulse are controlled where possible to obtain reliable spectra, and those measures are discussed in detailed in chapter 2. Unknowns relating to the properties of the tablet, such as the thermal conductivity, hardness, and API homogeneity, are not addressed as this work was concerned with applying techniques to off-the-shelf products as opposed to samples made in a lab where potential differences between pharmaceutical tablets are eliminated. These unknown properties will be considered as intrinsic to each tablet and will all contribute to the final spectrum that is produced in a LIBS experiment. The limitation of the spectrometers field of view forms the basis for the concept of spatial resolution, discussed in section 1.4.

Methodologies have been outlined in the literature to work with pharmaceuticals. By increasing the number of sites on the pharmaceutical, as shown in figure 1.1, as well as taking into consideration the geometry of the tablet and the coating processes, one can better study variations in tablet-to-tablet results and the overall results of a batch.

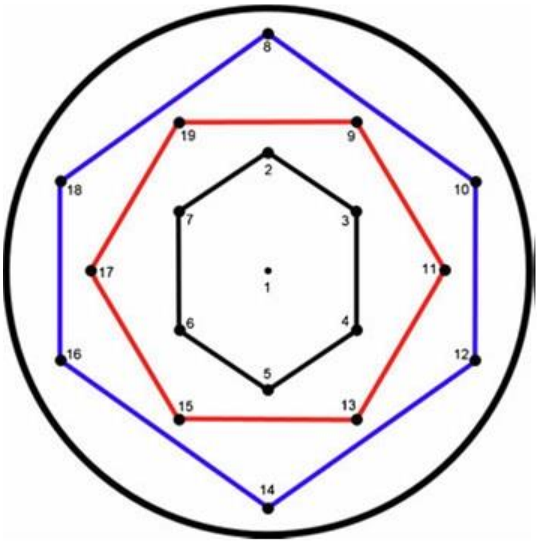


Figure 1.1: Schematic description of site distribution on LIBS-analysed tablets [82]. Each numbered point corresponds to a plasma formation site for LIBS, i.e. where the tablet is sampled. The spectra at each site can be compared to determine the uniformity of the tablet or tablet coating. As the tablet is curved, with the thickest point at the centre (site 1), the sites linked by coloured lines are formed under the same conditions, whereas the conditions may vary between sites located on different coloured lines.

Using a set system of sites, non-uniformities can be averaged out by analysing different areas of a pharmaceutical. An analysis of thicknesses at these sites can also be done to compare coating thickness across the various sites [91,92].

An experimental limitation of LIBS is that when analysing a curved tablet, a spectrum taken on the curved edges causes the laser optical path to intersect diagonally with the target, leading to an observed inconsistency over the target, while possibly also obscuring the plasma from the detector [92]. This presents a significant issue requiring the implementation of a probing methodology which acquires spectra on a flat face, if one exists, to obtain reliable spectra. If a flat face is not present spectra should only be recorded over a small region (i.e. the sampling region can be taken to be approximately flat) to prevent large variations dominated by curvature effects.

1.3 General characteristics of laser produced plasmas

As mentioned previously, LIBS is in principle a simple technique. A laser pulse is used to form a plasma on a sample and by detecting the emission produced information relating to the sample can be obtained. However, the origins of this emission, and its variation from sample to sample are not so simple. The following sections aim to provide an introduction to the origin of the some of the emission variation. This material is relevant as in my own work a laser pulse is incident onto a pharmaceutical tablet, which is a complex target, to form a plasma.

1.3.1 Laser plasma formation

When a laser pulse of sufficient energy is focused onto a solid target, such as a pharmaceutical tablet, a high-density high-temperature plasma is formed on the target surface. In essence a portion of the tablet near the surface is vaporised into a cloud of atoms and ions of various charges that can emit and absorb photons. A basic diagram of this is shown in figure 1.2. In the creation of a plasma by laser light the first stage to be considered is the heating of the surface of the material, mainly carried out by multiphoton ionisation (section 1.3.2.4) and/or by the interaction of the laser pulse with the electrons present in the material (i.e. inverse bremsstrahlung, section 1.3.2.1). The phase transition of the surface, from solid to liquid, is important as there is a change in the refractive index of the material, which leads to an increase in the rate of heating as the pulse energy is redistributed via electron collisions [93].

When the boiling point of the material is reached melting of the sample and vaporisation occurs. Melt zones observed on targets are melted and re-condensed matter [37]. As this region may no longer be indicative of the sample prior to being irradiated, choosing not to sample these melt zones can be a point of experimental optimisation depending on the complexity and homogeneity of the sample under investigation. At high intensities the evaporation rate is large and a vapour forms in front of the surface of the material. The vapour continues to grow, and it may have sufficient density to further increase the rate of absorption of the incident laser light. When evaporation occurs the ablation pressure increases, due to the recoil on the surface from the departing particles. The shock front due to this ablation pressure also travels into the material compressing and heating it further. As the temperature continues to increase a significant number of atoms will ionise due to collisions and this production of free electrons will increase the energy absorption rate from the laser beam, via electron-ion inverse bremsstrahlung absorption [94], i.e., the absorption of photons by free electrons resulting in their acceleration to higher energies. At this point a plasma is formed.

A plasma can be considered as a partially or fully ionised gas containing electrons, ions and neutral atoms. In contrast to neutral gases, the Coulomb forces between the charged particles in the plasma are strong and the range depends on the electron density [95, 96]. Hence the particles in the plasma interact with one another and tend to respond collectively [37]. The number density of electrons, n_e, is given by equation 1.1, where n_z is the number density of ions of charge Z.

$$n_e = \sum_z n_z Z \tag{1.1}$$

This equation describes the plasma as being overall neutral with the number of electrons equalling the sum of the number of ions multiplied by their charge state. This means there are more electrons when there are higher charge state ions, as those ions have lost more electrons to get to those higher charge states. When the plasma cools the number of electrons goes down as the ions trapping them go to lower charge states [97, 98].

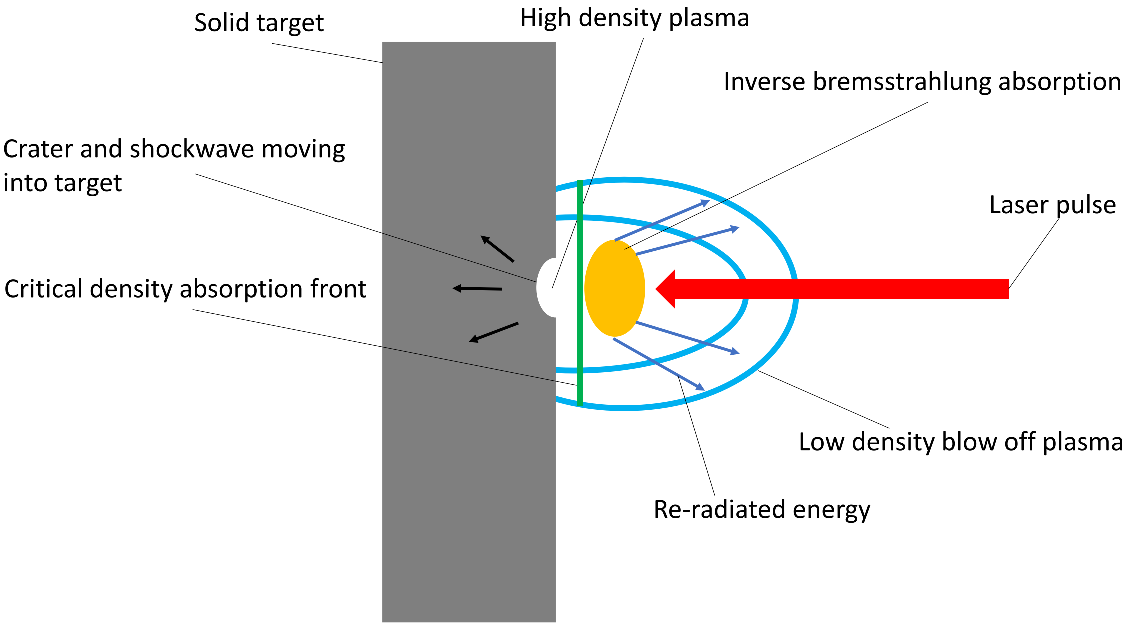


Figure 1.2: Diagram of a laser plasma formation, showing some of the key regions.

If there is sufficient energy to produce ionisation then there will also be excitation, when these excited states decay, light can be produced. In general, the spectrum of a laser produced plasma will consist of lines and continua. The relative contributions from line radiation, recombination and bremsstrahlung in a given spectrum are heavily dependent on the power density of the incident laser pulse, the target material and the spectral region being observed. A plasma is generally characterised by its electron or ion density and a measure of the energy of the charged particles, often given in terms of the temperature of the electrons present in the plasma.

Once a plasma is present, the dominant form of absorption of the laser pulse energy is by inverse bremsstrahlung within the plasma, in which an electron in a binary collision with an ion, absorbs a photon [93]. This quickly leads to higher plasma temperatures and increased ionisation.

In laser produced plasmas there is generally a higher density close to the ablation site, an absorption front where the plasma frequency and laser frequency has coupled to produce the critical density and an optically opaque zone between the region at the critical density and the target surface. As the plasma expands it cools and the density drops becoming optically thin again, allowing the laser radiation to propagate and be absorbed by the plasma via inverse bremsstrahlung once more. The exterior of the plasma now consists of low-density plasma. As the laser pulse continues to excite the plasma/target, the cycle continues: generating a critical density in the plasma with shielding, before the density drops to become optically thin again and transmissive so inverse bremsstrahlung absorption can occur again.

1.3.2 Excitation and de-excitation processes

In a plasma there are three main processes responsible for the emission of radiation: bound-bound, free-bound and free-free transitions. Each of these have a number of mechanisms associated with them, both collisional and radiative. Collisional processes dominate at high plasma densities, while radiative processes dominate at low plasma densities. Plasma density, electron temperature and the competing collisional and radiative processes are key factors in the determination of the energy partitioning in the plasma and the ionisation balance within the plasma. Energy transfer within a plasma can be accounted for by these radiative and collisional interactions. Those processes and inverse processes, as laid out in table 1.1, will be described below in the following sections [99].

Table 1.1: Summary of atomic processes in laser induced plasmas.

Process	**Excitation**	**De-excitation**	**Type**
F↔F	Bremsstrahlung	-	Collisional
F↔F	-	Inverse bremsstrahlung	Radiative
F↔B	Impact ionisation	3-Body recombination	Collisional
F↔B	Photoionisation	Radiative recombination	Radiative
B↔B	Impact excitation	Impact de-excitation	Collisional
B↔B	Photoabsorption	Spontaneous decay	Radiative

1.3.2.1 Free-free transitions

Free-free transitions involve the collision of a free electron with an ion, resulting in the transition of the electron to an unbound state of lower energy, with the emission of a photon. The radiation emitted is referred to as bremsstrahlung and the spectra produced by this process are continua [100]. Inverse bremsstrahlung involves the absorption of a photon by the electron-ion system, causing the free electron to make a transition to a higher energy state in the free electron continuum. This process is considered to be the main mechanism of laser heating and ionisation in the formation of laser plasmas, namely the main source of energetic electrons which ionise the target when the laser pulse is incident on the target material [97, 98]. These processes are shown in equation 1.2, where X is the ion, e^{**} is the higher energy free electron, e^* the lower energy free electron and γ the emitted/absorbed photon.

$$X + e^{**} \rightleftharpoons X + e^{*} + \gamma \qquad (1.2).$$

Bremsstrahlung radiation occurs mainly during the early stages of a plasmas lifetime and close to the targets surface where the density is high.

1.3.2.2 Free-bound transitions

Free-bound emission transitions involve free electron capture by an (n + 1)-times ionised atom, resulting in a transition to an n-times ionised atom, the surplus energy being emitted as a photon, known as recombination radiation. For a given bound state, the emission spectrum is a continuum with a sharp low-frequency cut-off, known as the recombination limit, which corresponds to the minimum required energy for ionisation of the atom from that bound state. For absorption transitions the reverse of this may occur, known as is photoionisation, where an absorbed photon moves a bound electron into the continuum. There processes are shown in equation 1.3, X^{n} and X^{n+1} are ions n- and (n+1)-times ionised, e^{*} is a free electron and γ is an emitted/absorbed photon:

$$X^{n} + \gamma \rightleftharpoons X^{n-1} + e^{*} \qquad (1.3).$$

Three-body recombination is also possible, becoming more probable as the density rises, whereby two free electrons collide with the ion simultaneously, one electron recombines with the ion while the other electron carries the surplus energy released as kinetic energy. Thus recombination occurs without the emission of radiation. The inverse of this process is electron impact ionisation, which involves the collision of a free electron with an ion, resulting in the ejection of a bound electron into the continuum. There processes are shown in equation 1.4, X^{n} and X^{n-1} are ions n- and (n-1)-times ionised and e^{***}, e^{**} and e^{*} are free electrons of decreasing energy:

$$X^{n} + e^{***} \rightleftharpoons X^{n-1} + e^{**} + e^{*} \qquad (1.4).$$

Free bound processes typically occur at intermediate times in the plasma lifetime when the plasma is in transition from continuum dominated radiation to line dominated radiation [97, 98].

1.3.2.3 Bound-bound transitions

Bound-bound transitions involve a transition within an ion stage from one bound state to another and are responsible for the unique discrete line emission from the various atomic species. Excitation of the ion may have occurred due to collision with other particles (collisional excitation) or absorption of a photon (photoexcitation). Emission may accompany the transition to a bound state of lower energy, giving rise to a spectral line, whose profile is dependent on the spontaneous lifetime of the upper state, and whose wavelength is ion specific. These processes are shown in equation 1.6, where X is an ion of less energy than X^* and γ is an emitted/absorbed photon:

$$X + \gamma \rightleftharpoons X^* \qquad (1.6).$$

The intensity of emission from this process is given by

$$I = \frac{hcN_0 gA}{4\pi\lambda Z} e^{\frac{-E}{kT}} \qquad (1.5),$$

where N_0 is the total species population, Z is the partition function g is the statistical weight of the ground state, A is the Einstein A coefficient, h is Planck’s constant, c is the speed of light, λ is the wavelength of the transition, E is the energy difference between the upper and lower levels and k is Boltzmann’s constant and T is the temperature [37].

Typically, an emission line would have a Voigt profile. A Voigt profile is the convolution of a Gaussian and Lorenz profile, where there are physical phenomena behind each of these contributing profiles. For example, a Gaussian line profile is the result of Doppler broadening. Doppler broadening is line broadening due to the Doppler effect, which is caused by the different velocity of particles. Collisional broadening, due to collisions with neutrals, ions and electrons in the plasma, results in a Lorenz line profile. Broadening due to the lifetime of excited states (Lorenz) and instrumental broadening due to the spectrometer and detector (Gaussian or Lorenz) used will also be present.

As both Gaussian and Lorenz components (often of comparable magnitude) are present the final recorded spectrum the recorded emission line will have a Voigt profile. At the line centre the Gaussian profile usually dominates, while at the wings of emission lines the Lorentz profile often dominates [37]. As the underlying parameters in the plasma change with time so too will the observed parameters of the Voigt profile for an observed line.

Collisional excitation involves a collision between an electron and an atom/ion resulting in the transfer of kinetic energy to excitation energy in the atomic system. Collisional de-excitation is the inverse process, with both shown in equation 1.7 where X and X' denote two atomic species colliding with the * indicating the one with more energy [97,98].:

$$X^* + X' \rightleftharpoons X + X'^* \qquad (1.7).$$

Bound-bound transitions mostly occur when the bulk of the plasma has cooled significantly from the initial formation.

1.3.2.4 Multiphoton process

In the description of free-bound and bound-bound transitions single photon interaction were assumed. However, these processes can also take place even when single photons, individually, do not have the necessary energy. Multiphoton processes may occur when the cumulative energy of multiple absorbed photons is sufficient to excite electrons and free them (photoionisation), or to excite electrons to higher energy levels (photoabsorption) [101].

1.3.3 Thermodynamic equilibrium in a plasma

A plasma is said to be in the steady state when its rate of ionisation and the rate of recombination are equal. The ratio of the density of one ionised species to another, the ionisation equilibrium, must be known in order to describe the state of a plasma. It is also necessary to consider the time required to reach this steady state, which generally varies inversely with the electron number density. If a system is in thermodynamic equilibrium, then temperature can be used to describe plasma properties like the distribution of the speed of particles and the relative populations of energy levels.

There are three standard models for ionisation equilibrium: local thermodynamic equilibrium (LTE), coronal equilibrium (CE) and collisional radiative equilibrium (CRE).

Full thermodynamic equilibrium is rarely complete so an approximation, LTE, is often used. The criterion for LTE is that collisional processes must be more important than radiative ones. Thus, an excited state must have a higher probability of de-excitation by collision than by spontaneous radiation. This approximation is only valid at sufficiently high densities, where the distribution of ion stages is determined only by collisional processes. The cut-off density given by

$$n_e \geq 1.6 \times 10^{12} {T_e}^{1/2} \chi^3 cm^{-3} \qquad (1.8),$$

where T_e is the electron temperature and χ is the excitation energy potential in eV of the corresponding transition or line emission [102].

If electron densities are too low for LTE to exist, another type of equilibrium model may be applied. One of these models is the CE model, in which it is assumed that transitions from the ground state, or low excited states, to higher excited states occur due to electron collisions, but radiative processes dominate the de-excitation and recombination processes [95]. Under CE conditions, the partial densities of the excited states do not obey a Boltzmann distribution, but rather the ground state is over-populated. Although excitation rates are low, the ion, once excited, has a negligible probability of decaying by any process other than spontaneous radiation [103]. This equilibrium model is applicable to low plasma densities and high temperatures, such as astrophysical plasmas as in the Sun's corona. An approximation of the required electron density to apply the CE model is:

$$n_e \geq 5.6 \times 10^8 (Z+1) {T_e}^{1/2} e^{\frac{1.162\times 10^3\,(Z+1)^2}{T_e}} \tag{1.9}$$

where T_e is the electron temperature and Z is the average charge in the plasma [101].

The CRE model is the model most appropriate to the laser produced plasmas in this work. The plasma observed is neither at the high-density case of LTE nor at the astrophysical low density appropriate for CE – the plasma has a balance of collisional and radiative processes. Also, as the CRE model is an intermediate model, the conditions of LTE and CE can be obtained using the CRE model with the extreme conditions of the other models.

The work of Colombant and Tonton [104] introduces a model to describe this regime, by balancing the collisional and radiative processes. In this model the ion density populations are is given by:

$$\frac{n_{z+1}}{n_z} = \frac{S(z,T_e)}{\alpha_r(z+1,T_e) + n_e\alpha_{3b}(z+2,T_e)} \quad (1.10),$$

where, $n_e$is the electron density, $S(z,T_e)$ is the collisional ionisation coefficient, $\alpha_r(z,T_e)$ is the radiative recombination coefficient and $\alpha_{3b}(z,T_e)$ is the three-body coefficient recombination, each of which are functions of the charge state (z), electron temperature (T_e), ionisation potential (χ_z) and the number of open shell electrons (ξ_z). $S(z,T_e)$, $\alpha_r(z,T_e)$ and $\alpha_{3b}(z,T_e)$ are given by equation 1.11–1.13, respectively.

$$S(z,T_e) = \frac{9 \times 10^{-6}\xi_z\sqrt{\frac{T_e}{\chi_z}}}{\chi_z{}^{\frac{3}{2}}\left(4.88\frac{T_e}{\chi_z}\right)} e^{\left(\frac{-\chi_z}{T_e}\right)} \ cm^3 s^{-1} \quad (1.11),$$

$$\alpha_r(z,T_e) = 5.2 \times 10^{-14}\left(\sqrt{\frac{\chi_z}{T_e}}\right) Z\left[0.429 + \frac{1}{2}\ log\left(\frac{\chi_z}{T_e}\right) + 0.469\sqrt{\frac{T_e}{\chi_z}}\right] cm^3 s^{-1} \quad (1.12),$$

$$\alpha_{3b}(z,T_e) = 2.9 \times 10^{-27}\frac{\xi_z}{T_e\chi_z}\left(4.88 + \frac{T_e}{\chi_z}\right) s^{-1} \quad (1.13).$$

The state of charge densities is temperature dependent, meaning that different charge states become more or less populated at different temperatures. While line analysis can be used in LIBS studies to determine experimental parameters such as plasma temperature, those studies require a time resolved experimental setup, whereas the experimental setup used in this work, and described in detail in chapter 2, is time integrated.

1.4 Time-integrated spatially resolved LIBS

From the discussion in the previous sections, it is clear that plasmas are complicated, with spatially and temporally varying properties that will affect the emission. Indeed, detailed models can be very complex requiring super-computers [105, 106]. However, some useful insight can be gained, even from the brief descriptions given above. Firstly, the hottest most dense part of the plasma exits in the early stages of its lifetime, when the laser pulse is still on and the denser portions of the plasma are located close to the surface of the target. Secondly, at later times and as the plasma expands the temperature and density of the plasma begins to decay. This evolution of the plasma can be taken advantage of to improve the performance of LIBS.

The most common configuration of LIBS is time-resolved laser-induced breakdown spectroscopy (TRLIBS), incorporating high quality, sensitive gated image detectors, such as intensified charge coupled devices [37]. In this configuration the spectrum is not recorded until sometime, Δt (ns to μs), after the plasma is formed, and recorded for a specific duration (from ns to ms). This allows a reduction in the level of background continuum emission detected from the early stages of the plasma lifetime, while particular transitions from specific ion stages can be isolated in time leading to higher sensitivity.

However, applying TRLIBS to the VUV is more difficult, due to the relatively poor detector options available to give the required spectral and temporal coverage. The system in DCU instead uses time-integrated spatially resolved (TISR) LIBS, which is based on the spatial, rather than temporal, characteristics of the plasma evolution (see figure 1.3).

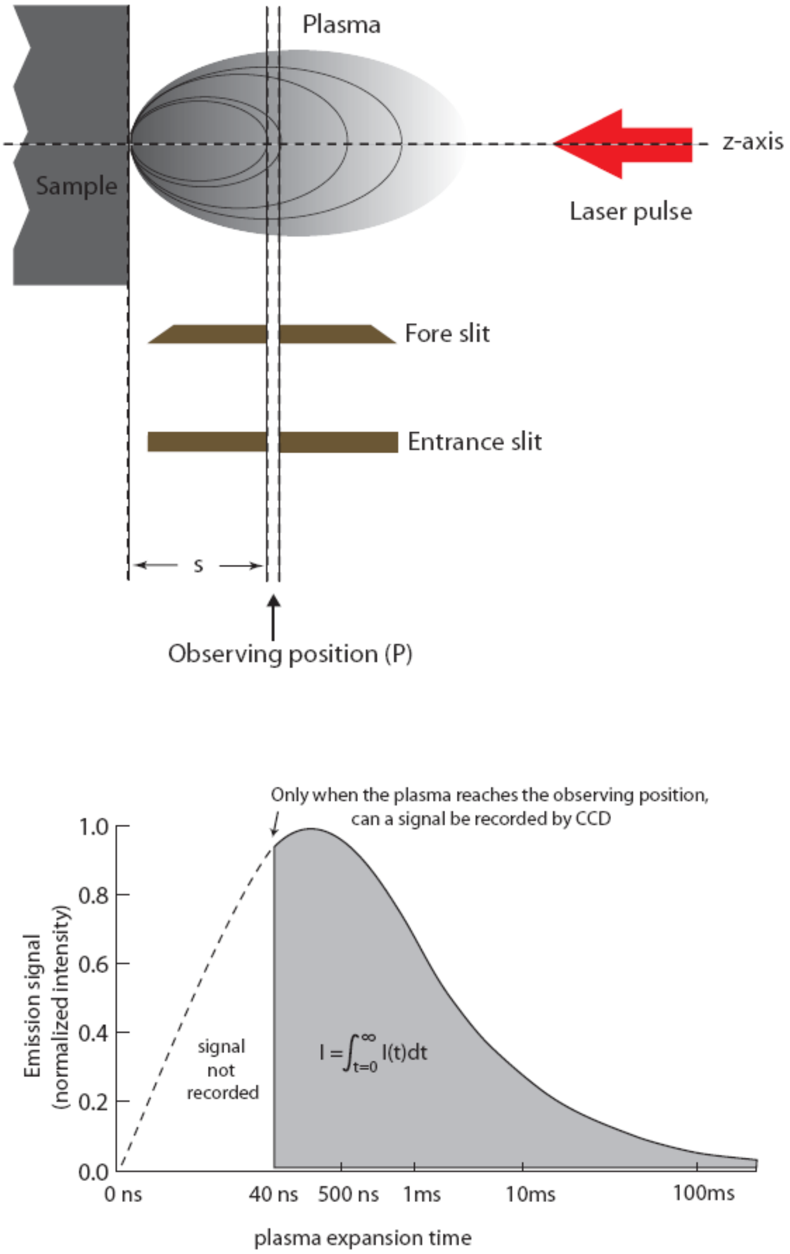


Figure 1.3: Schematic diagram of the TISR scheme [97], where the plasma is allowed to expand and cool before it becomes visible to an observer (e.g., a spectrometer).

The plasma is allowed to expand out from the target surface, so that the continuum emission is replaced by line emission as the electron density and temperature in the plasma drop off. This means that radiation from the entire time duration of the plasma can be captured if the hot dense zone close to the target surface is shielded from the spectrometer. This can be achieved using a fore-slit in conjunction with the entrance slit to a spectrometer. This double slit arrangement, if aligned correctly, will limit the field of view of the spectrometer. A position can be chosen such that the line emission is still strong but is far enough away from the target surface that the continuum emission is low.

1.5 Use of multivariate techniques in LIBS

Sections 1.3 and 1.4 gave a brief introduction to the properties of laser produced plasmas and the observational techniques used to generate spectra which are used for LIBS. Typically, statistical analysis in a study would first focus on univariate analysis, the analysis of a single variable or parameter, before escalating to multivariate analysis. In some cases, such as trying to determine low concentrations of carbon in a steel samples, this is appropriate [97]. As the carbon constitutes less than 1% of the sample it is safe to assume that different samples are effectively identical, and the changes observed in the carbon emission features are only due to slight change in carbon concentration once the experimental conditions remain constant.

However, in the more general case, the processes require to produce emission for LIBS is dependent on the physical parameters of the sample and the observational techniques in use. The composition, structure, hardness, colour, thermal properties, chemical properties, shape and surface quality, amongst other properties, of the surface will all contribute to the initial laser-target interactions that start the process leading to emission spectra. As one investigates different samples, or different sites of inhomogeneous samples, some (if not all) of these properties will change, leading to different compositions and differing amounts of laser energy remaining after ablation to drive processes within the plasma. As the rates for the various excitation and de-excitation process in a plasma depend on the ions present, the electron density present and electron temperature, a knock-on effect will be present in these also.

All of these variations will lead to changes in ion density populations, given by equation 1.10, and in turn the intensity of the line emission (equation 1.5). Therefore the emission from the plasma at all times and points in space will be determined by the properties of the target, laser pulse properties, plasma physics and atomic physics, with an inter relationship between all.

Even in an ideal case it would not be expected that fine control over the production of pharmaceutical samples for a LIBS study, where a property is modified in a known measured way, would lead to a simple relationship to the recorded spectra as the emission is not generally determined by an individual physical parameter. Previous work has, however shown that multivariate analysis is suitable in LIBS applications [86].

Multivariate analysis allows one to investigate how a whole spectrum, or a number of components of a spectrum), changes from sample to sample, or position to position and how that may be used to group related spectra into individual classes, i.e., classification of spectra arising from say different target materials into different groups or classes. Multivariate analysis can take two forms. The first is where a number of parameters, say the colour, shape and composition of a tablet, are measured and then a relationship to the recorded spectra is determined. In other words, the variables effecting the spectra are defined and the coefficients to explain observed variance are found. The second approach, and the one used in this work, is instead to assume that variations are present in the samples and these unknown variables will lead to changes in the recorded spectra. The goal is then to find a mathematical representation of these variables (which might have no direct or meaningful physical equivalence) and to again determine coefficients to explain observed variance are found. This does mean, however, that the approach taken in this thesis is a whole spectrum form of analysis as the information for these unknown variables is contained in the entire spectrum rather than just a single line.

Using multivariate analysis, such as PCA described in section 1.5.1, the arduous experimental optimization to enhance the line emission and reduce the continuum emission, resulting in optimal signal-to-background ratios and hence LOD, is no longer the focus of the LIBS experiment. In this case one is interested in how a spectrum changes rather than the detailed spectral shape. A number of these techniques are developed and implemented in this thesis to study emission from plasmas formed from pharmaceutical samples.

This whole spectrum approach can be taken one step further. Instead of looking for the underlining variables (either known or unknown), machine learning can be utilised to recognise complex patterns in the spectra. In effect the causes of the spectral variations lose importance, but the net spectral changes take primacy. The goal is then to teach the machine learning algorithm, or allow them to self-learn, to associate different spectra with different samples and to test to see how it well it correctly classifies previously unseen spectra. An overview of machine learning in LIBS is given in section 1.5.2, while details of the approaches I used are in chapter 3.

1.5.1 Principal component analysis

Throughout the literature several numerical methods have been presented for consideration, such as partial least-squares discriminant analysis and soft independent modelling of class analogy [86, 107]. After reviewing the relevant literature PCA was chosen as it is relatively easy to implement, and it presented a solution for quick visual inspection that gave an indication of the similarity/dissimilarity of spectra. It also provides a means to achieve dimensional reduction. [86, 93].

Put simply PCA is a chemometric technique that allows one to group or cluster spectra based on similarity. PCA can be, and has been, used to tell pharmaceutical analytes apart in conjunction with LIBS [108, 109]. In terms of practical visualisation, similar analytes group together, and dissimilar analytes appear further away in space.

The implementation of PCA can be summarised as follows. Let A be the input matrix, where each row is a recorded spectrum (i.e., intensity values for each wavelength). The covariance matrix is generated by multiplying the transpose of A by A, i.e., $A\times(A^{T})$. The eigenvectors are designated by V, obtained via eigen-decomposition and sorted by eigenvalues of the covariance matrix. By sorting eigenvectors by eigenvalues, the eigenvectors that h T_k, given by

$$T_k=A\times V_k \tag{1.8}$$

where V_k contains the selected k columns from V, is then the principal component (PC) score and these values are used as co-ordinates to plot points, each point representing one of the input spectra. For the studies presented in this thesis the first three PC scores are monitored (k=3). In a similar pharmaceutical LIBS study the first two PCs contained 90.67% of the variance in the dataset, while the first three PCs contain 96.29% of the dataset [86], indicating that by incorporating the third PC more information regarding the variance of the analytes is preserved, as compared to just using the first two PCs.

Let us take a simple example. Take 5 spectra, each of 4 pixels with each pixel value assigned a random number from 0 to 1, as seen in table 1.2. By applying equation 1.14, we can calculate the variance preserved by each PC. Since the eigenvectors are sorted by eigenvalues the first PC holds the most variance, followed by the second and so on, as observed in table 1.3. Note that the sum of all PCs is 100%. In this case retaining the first two PCs would retain >92% of the total variance, while the first three would retain >98% of the total variance. All the variance could be retained by keeping all four PCs, but the size of the output in that case would be the same as the input and a simple visualisation would not be possible.

$$\text{Variance preserved by a PC} = \frac{\text{Eigenvalue of interest}}{(\sum|\text{All Eigevalues}|) \times 100} \qquad (1.14),$$

Table 1.2: Five example input spectra, each 4 pixels long.

	Pixel 1	Pixel 2	Pixel 3	Pixel 4
Spectrum 1	0.4173	0.4893	0.7803	0.1320
Spectrum 2	0.0497	0.3377	0.3897	0.9421
Spectrum 3	0.9027	0.9001	0.2417	0.9561
Spectrum 4	0.9448	0.3692	0.4039	0.5752
Spectrum 5	0.4909	0.1112	0.0965	0.0598

Table 1.3: Variance preserved by each PC and the sum of all PCs.

	% Variance Preserved
PC1	82.9
PC2	9.3
PC3	6.3
PC4	1.5
Sum	100

Hence PCA, as applied here, is in effect dimensional reduction. By manipulating tensors via simple multiplication, it is possible to reduce a long vector spectrum into three coordinates, an easy to plot and interpret X, Y and Z entity, but with the majority of the variance retained.

Now that simple 3-D plots of these points can be made there is a need for a means to classify them. As the PCA takes care of clustering them, a simple geometric volume or shape can be used to create a decision boundary. If the points are within the shape they belong to a certain class, if not they do not. The shape chosen was the ellipsoid [110], where the centre points are the means of the data points in their respective directions and the semi axis length in each direction corresponds to three times the standard deviation of the PC score in that direction. Equation 1.15 shows the formula used to generate the data points for the ellipsoid.

$$\frac{(x-xc)^2}{xr^2}+\frac{(y-yc)^2}{yr^2}+\frac{(z-zc)^2}{zr^2}=1 \quad (1.15),$$

where xc is the centre point x coordinate, yc is the centre point y coordinate and zc is the centre point z coordinate. xr is the semi axis length of the x axis, yr is the semi axis length of the y axis, zr is the semi axis length of the z axis. x, y and z are the data points being calculated. Figure 1.4 demonstrates such a visualisation with PCA and an ellipsoid classifier.

Note that if all the semi axis lengths are set to the same values the result is a sphere. The reason an ellipsoid is chosen is that the volume of an ellipsoid can enclose a dataset more closely than a sphere. If less distribution is observed about one or more of the axes, the volume about that axis can be shrunk to better represent a volume a particular sample would occupy if it belonged to a certain class.

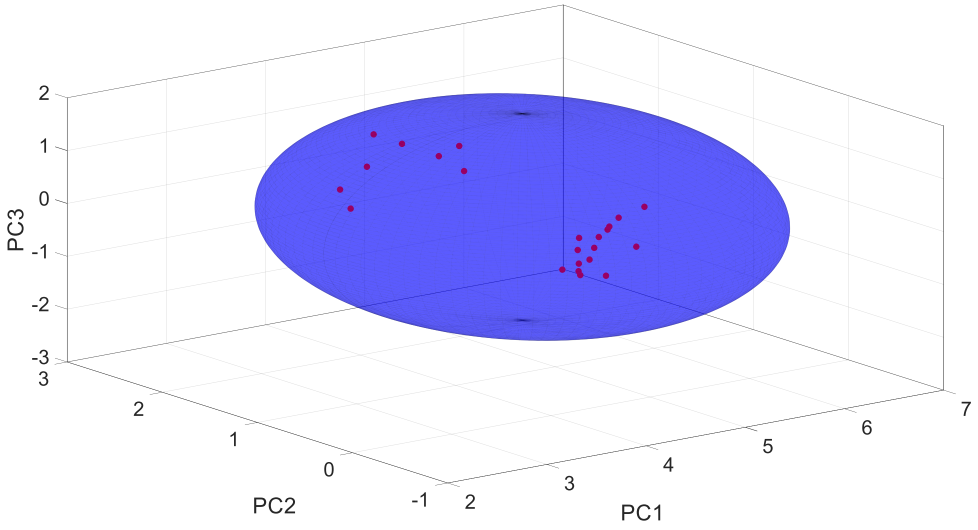


Figure 1.4: Visualization of an ellipsoid classifier, dimensionally reduced spectra are represented here as red points, each one corresponding to an input spectrum.

1.5.2 Machine learning

Several machine learning approaches have been applied to LIBS, and some of these techniques will be described in detail in chapter 3. A combination of artificial neural networks and PCA with LIBS has been used to determine the location of origin of herbal medicines [111]. Support vector machines (SVM) have been used with LIBS to distinguish pharmaceutical analytes [112]. SVMs, with LIBS, have also been shown to be capable of telling jade stones apart [113], identifying coffee varieties [114], shown capable of identifying and sorting PVC polymers in the recycling process [115], discriminating between different soft tissues such as fat skin and muscle [116] and estimating the aging grade of steel [117].

Self-organizing maps (SOM) have been used in LIBS to identify industrial polymers [118], analyse waste electrical equipment [119] and for various archaeological studies [120-124]. SOMs have demonstrated their capability for fast quantitative elemental mapping of samples [125].

A study by Chengxu et al. demonstrated that convolutional neural networks (CNNs), used in combination with LIBS, were capable of determining the concentration of potassium in soil [126]. Chengxu et al. used time resolved spectra, took the spectra from the first 20 times and applied PCA with K=20 to generate an image as the input for a CNN. The lithological study by Chen et al. used CNNs in conjunction with LIBS for the classification of minerals. Many spectra that constituted a wavelength scan were reshaped to make an input image for a CNN [127].

It should be noted that the work in this thesis is qualitative not quantitative, but it is believed to be the first of its kind in pharmaceutical LIBS to convert an individual spectrum into an image to be used as an input to a CNN.

1.6 Summary

In this chapter the aims and objectives of the work have been outlined. My contributions to the work have also been outlined. This work has been defined as a proof of principle which is tested in chapter 5. LIBS has been introduced and a literature review focusing on its pharmaceutical applications has been outline. Relevant material on laser produced plasmas has been detailed. Time integrated spatially resolved LIBS has been covered and forms the basis of the experimental application, the experimental setup is shown in chapter 2. The use of multivariate techniques as well as a full spectrum approach to analysing pharmaceuticals has been outlined, with PCA being the multivariate technique selected for use in this thesis. Literature supporting the application of machine learning in LIBS has been presented, these techniques are further explained in chapter 3.

References

[1] The European Medicines Agency. Quality by design [Internet].2011 [Cited 28/12/2019]. Available from: https://www.ema.europa.eu/en/human-regulatory/research-development/quality-design

[2] Sainz V, Conniot J, Matos AI, Peres C, Zupanŏiŏ E, Moura L, Silva LC, Florindo HF, Gaspar RS. Regulatory aspects on nanomedicines. Biochemical and biophysical research communications. 2015 Dec 18;468(3):504-10.

[3] Pharmachemical Ireland. Ireland Strategy in Action. 2012; pharmachemicalireland.ie/Sectors /PCI/PCI.nsf/vPages/PCI_policy~Publications_and_Resources~strategy-in-action.

[4] EI and IDA. Call for Pharmaceutical Manufacturing Technology Centre (PMTC). 2012; enterprise-ireland.com/technologycentres.

[5] FORAS. Report of the Research Prioritisation Steering Group. 2011; http://www.forfas.ie/ media/ffs20120301-Research_Prioritisation_Exercise_Report.pdf.

[6] FDA. Guidance for industry PAT. 2004; www.fda.gov/downloads/Drugs/GuidanceCompliance RegulatoryInformation/Guidances/ucm070305.pdf.

[7] European Federation of Pharmaceutical Industries and Associations. Outline Strategic Research Agenda for a biomedical research public private partnership under Horizon 2020. 2013; efpia.eu/ uploads/Modules/MCMedias/1373296554546/IMI2 Strategic_Research_Agenda_v 8 July 2013.pdf.

[8] Gaspar R, Aksu B, Cuine A, Danhof M, Takac MJM, Linden HH, et al. Towards a European Strategy for Medicines Research (2014-2020): The EUFEPS Position Paper on Horizon 2020. Eur J Pharm Sci 2012;47(5):979-987.

[9] Hutcheson JD, Goergen CJ, Schoen FJ, Aikawa M, Zilla P, Aikawa E, Gaudette GR. After 50 Years of Heart Transplants: What Does the Next 50 Years Hold for Cardiovascular Medicine? A Perspective from the International Society for Applied Cardiovascular Biology. Frontiers in cardiovascular medicine. 2019;6.

[10] Peruzzi M, Zoccai GB, Frati L, De Falco E, Chimenti I, Greco E, Marullo AG, Frati G, Vitulli P. Bridging regenerative medicine based therapies into the 21st Century: solo or symphony? International Archives of Medicine. 2015 May 25;8.

[11] Davoodi S, Kariminejad R, Mohammadzadeh Z. Cancer care improvement through positive patient identification technologies. International Journal. 2018 Sep;4(9):215.

[12] Páez-Avilés C, Juanola-Feliu E, Tahirbegi IB, Mir M, Samitier J. CoPs in innovation and technology transfer of medical devices. InISPIM Conference Proceedings 2015 (p. 1). The International Society for Professional Innovation Management (ISPIM).

[13] Decano JL, Aikawa M. Dynamic macrophages: understanding mechanisms of activation as guide to therapy for atherosclerotic vascular disease. Frontiers in cardiovascular medicine. 2018;5.

[14] Páez-Avilés C, Juanola-Feliu E, Bogachan-Tahirbegi I, Mir M, González-Piñero M, Samitier J. Innovation and technology transfer of medical devices fostered by cross-disciplinary communities of practitioners. International Journal of Innovation Management. 2015 Dec 15;19(06):1540012.

[15] Rao M. National Institutes of Health Center for Regenerative Medicine: putting science into practice. Stem cells and development. 2013 Dec 1;22(S1):4-7.

[16] Bagley CE, Tvarno CD. Pharmaceutical public-private partnerships: Moving from the bench to the bedside. Harv. Bus. L. Rev.. 2014;4:373.

[17] Demetzos C, Pippa N. Advanced drug delivery nanosystems (aDDnSs): a mini-review. Drug Delivery. 2014 Jun 1;21(4):250-7.

[18] Sanchez-Jimenez F, Reyes-Palomares A, A Moya-Garcia A, AG Ranea J, Angel Medina M. Biocomputational resources useful for drug discovery against compartmentalized targets. Current pharmaceutical design. 2014 Jan 1;20(2):293-300.

[19] Meijerman I, Nab J, Koster AS. Designing and implementing an inquiry-based undergraduate curriculum in pharmaceutical sciences. Currents in Pharmacy Teaching and Learning. 2016 Nov 1;8(6):905-19.

[20] Siiskonen MD. Integrated Product and Production Platforms for Pharmaceutical Products: Design Thinking for the Development of Personalized Medicines (Doctoral dissertation, Chalmers Tekniska Hogskola (Sweden)); 2019.

[21] Steinmetz F. Integration of Data Quality, Kinetics and Mechanistic Modelling into Toxicological Assessment of Cosmetic Ingredients (Doctoral dissertation, Liverpool John Moores University); 2016.

[22] Farjadian F, Ghasemi A, Gohari O, Roointan A, Karimi M, Hamblin MR. Nanopharmaceuticals and nanomedicines currently on the market: challenges and opportunities. Nanomedicine. 2018 Nov 19;14(1):93-126.

[23] Duncan R. Polymer therapeutics: top 10 selling pharmaceuticals—what next?. Journal of Controlled release. 2014 Sep 28;190:371-80.

[24] Fonteyne M, Vercruysse J, De Leersnyder F, Van Snick B, Vervaet C, Remon JP, De Beer T. Process Analytical Technology for continuous manufacturing of solid-dosage forms. TrAC Trends in Analytical Chemistry. 2015 Apr 1;67:159-66

[25] Demetzos C. The Innovations in Science and Technology as a Demand for Bio-better Medicines in Europe. InGeNeDis 2014 2015 (pp. 203-208). Springer, Cham.

[26] Ganoth A, Tsfadia Y, Wiener R. Ubiquitin: molecular modeling and simulations. Journal of Molecular Graphics and Modelling. 2013 Nov 1;46:29-40.

[27] Bakshi S, de Lange EC, van der Graaf PH, Danhof M, Peletier LA. Understanding the behavior of systems pharmacology models using mathematical analysis of differential equations: prolactin modeling as a case study. CPT: pharmacometrics & systems pharmacology. 2016 Jul 1;5(7):339-51.

[28] Kurniawan KH, Pardede M, Hedwig R, Abdulmadjid SN, Lahna K, Idris N, Jobiliong E, Suyanto H, Suliyanti MM, Tjia MO, Lie TJ. Practical and highly sensitive elemental analysis for aqueous samples containing metal impurities employing electrodeposition on indium-tin oxide film samples and laser-induced shock wave plasma in low-pressure helium gas. Applied optics. 2015 Sep 1;54(25):7592-7.

[29] Mowery MD, Sing R, Kirsch J, Razaghi A, Béchard S, Reed RA. Rapid at-line analysis of coating thickness and uniformity on tablets using laser induced breakdown spectroscopy. Journal of pharmaceutical and biomedical analysis. 2002 Jun 1;28(5):935-43.

[30] Contreras VU, Meneses-Nava MA, Ornelas-Soto N, Barbosa-García O, López-de-Alba PL, Maldonado JL, Ramos-Ortiz G, Acevedo-Aguilar FJ, López-Martínez L. Fast

and environmentally friendly quantitative analysis of active agents in anti-diabetic tablets by an alternative laser-induced breakdown spectroscopy (LIBS) method and comparison to a validated reversed-phase high-performance liquid chromatography (RP-HPLC) method. Applied spectroscopy. 2012 Nov;66(11):1294-301.

[31] St-Onge L, Kwong E, Sabsabi M, Vadas EB. Quantitative analysis of pharmaceutical products by laser-induced breakdown spectroscopy. Spectrochimica Acta Part B: Atomic Spectroscopy. 2002 Jul 31;57(7):1131-40.

[32] Yokoyama M, Tourigny M, Moroshima K, Suzuki J, Sakai M, Iwamoto K, Takeuchi H. A novel rapid quantitative analysis of drug migration on tablets using laser induced breakdown spectroscopy. Chemical and Pharmaceutical Bulletin. 2010 Nov 1;58(11):1521-4.

[33] Lednev VN, Pershin SM, Sdvizhenskii PA, Grishin MY, Fedorov AN, Bukin VV, Oshurko VB, Shchegolikhin AN. Combining Raman and laser induced breakdown spectroscopy by double pulse lasing. Analytical and bioanalytical chemistry. 2018 Jan 1;410(1):277-86.

[34] Bush L, Sabsabi M. LIBS: Fundamentals, benefits, and advice to new users. Spectroscopy. 2019 Apr; 34(4):48-51.

[35] Vankeirsbilck T, Vercauteren A, Baeyens W, Van der Weken G, Verpoort F, Vergote G, Remon JP. Applications of Raman spectroscopy in pharmaceutical analysis. TrAC trends in analytical chemistry. 2002 Dec 1;21(12):869-77.

[36] Noll R. Laser Induced Breakdown Spectroscopy Fundamentals and Applications Springer. 2012;

[37] Cremers DA, Radziemski LJ. Handbook of laser-induced breakdown spectroscopy. John Wiley & Sons; 2013 May 13.

[38] Jiang X, Hayden P, Laasch R, Costello JT, Kennedy ET. Inter-pulse delay optimization in dual-pulse laser induced breakdown vacuum ultraviolet spectroscopy of a steel sample in ambient gases at low pressure. Spectrochimica Acta Part B: Atomic Spectroscopy. 2013 Aug 1;86:66-74.

[39] Jiang X, Hayden P, Costello JT, Kennedy ET. Double-pulse laser induced breakdown spectroscopy with ambient gas in the vacuum ultraviolet: Optimization of

parameters for detection of carbon and sulfur in steel. Spectrochimica Acta Part B: Atomic Spectroscopy. 2014 Nov 1;101:106-13.

[40] International Organization for Standardization. ISO 21348 Definitions of Solar Irradiance Spectral Categories. [Internet]. 2007 [Cited 17/09/2020]. Available from: https://web.archive.org/web/20131029233428/http://www.spacewx.com/pdf/SET_21348_2004.pdf

[41] Kelly RL, Palumbo LJ. NRL Report 7599. Naval Research Laboratory, Washington, DC, USA. 1973;

[42] Radziemski L, Cremers D. A brief history of laser-induced breakdown spectroscopy: from the concept of atoms to LIBS 2012. Spectrochimica Acta Part B: Atomic Spectroscopy. 2013 Sep 1;87:3-10.

[43] Debras-Guédon J, Liodec N. De l'utilisation du faisceau d'un amplificateur a ondes lumineuses par émission induite de rayonnement (laser à rubis), comme source énergétique pour l'excitation des spectres d'émission des éléments. CR Acad. Sci. 1963;257:3336.

[44] Wisbrun R, Schechter I, Niessner R, Schroeder H, Kompa KL. Detector for trace elemental analysis of solid environmental samples by laser plasma spectroscopy. Analytical Chemistry. 1994 Sep 1;66(18):2964-75.

[45] D'Orazio M, Feigl P, Grix R., von Hoerner H, Krueger FR, Li G, Miner G, Rohr W, Tinschmann A, Wollnik H. Facility for Remote Analysis of Small Bodies F.R.A.S., Parts I and II (Von Hoerner & Sulger Electronics GmbH, Schwetzingen, 1986, pp. 23–273.

[46] Maurice S, Wiens RC, Saccoccio M, Barraclough B, Gasnault O, Forni O, Mangold N, Baratoux D, Bender S, Berger G, Bernardin J. The ChemCam instrument suite on the Mars Science Laboratory (MSL) rover: Science objectives and mast unit description. Space science reviews. 2012 Sep 1;170(1-4):95-166.

[47] Wiens RC, Maurice S, Barraclough B, Saccoccio M, Barkley WC, Bell JF, Bender S, Bernardin J, Blaney D, Blank J, Bouyé M. The ChemCam instrument suite on the Mars Science Laboratory (MSL) rover: Body unit and combined system tests. Space science reviews. 2012 Sep 1;170(1-4):167-227.

[48] Carter S, Fisher A, Garcia R, Gibson B, Lancaster S, Marshall J, Whiteside I. Atomic spectrometry update. Review of advances in the analysis of metals, chemicals and functional materials. Journal of Analytical Atomic Spectrometry. 2015;30(11):2249-94.

[49] Khater MA. Laser-induced breakdown spectroscopy for light elements detection in steel: State of the art. Spectrochimica Acta Part B: Atomic Spectroscopy. 2013 Mar 1;81:1-0.

[50] Sturm V, Erben B, Fleige R, Wirz W. Carbon analysis of steel using compact spectrometer and passively Q-switched laser for laser-induced breakdown spectroscopy. Optics Express. 2019 Dec 9;27(25):36855-63.

[51] Bulajic D, Cristoforetti G, Corsi M, Hidalgo M, Legnaioli S, Palleschi V, Salvetti A, Tognoni E, Green S, Bates D, Steiger A. Diagnostics of high-temperature steel pipes in industrial environment by laser-induced breakdown spectroscopy technique: the LIBSGRAIN project. Spectrochimica Acta Part B: Atomic Spectroscopy. 2002 Jul 31;57(7):1181-92.

[52] Aimoto M, Kondo H, Yamamura H, Toh T. Chemical and Physical Analysis-Inspection of the Defects of Steel by Using Laser Induced Breakdown Spectroscopy. Journal of the Iron and Steel Institute of Japan-Tetsu to Hagane. 2007;93(7):483-8.

[53] Kraushaar M, Noll R, Schmitz HU. Slag analysis with laser-induced breakdown spectrometry. Applied spectroscopy. 2003 Oct 1;57(10):1282-7.

[54] Sturm V, Fleige R, de Kanter M, Leitner R, Pilz K, Fischer D, Hubmer G, Noll R. Laser-induced breakdown spectroscopy for 24/7 automatic liquid slag analysis at a steel works. Analytical chemistry. 2014 Oct 7;86(19):9687-92.

[55] Gondal MA, Hussain T, Yamani ZH, Bakry AH. Study of hazardous metals in iron slag waste using laser induced breakdown spectroscopy. Journal of Environmental Science and Health, Part A. 2007 May 7;42(6):767-75.

[56] Lorenzetti G, Legnaioli S, Grifoni E, Pagnotta S, Palleschi V. Laser-based continuous monitoring and resolution of steel grades in sequence casting machines. Spectrochimica Acta Part B: Atomic Spectroscopy. 2015 Oct 1;112:1-5.

[57] Cabalin LM, Delgado T, Ruiz J, Mier D, Laserna JJ. Stand-off laser-induced breakdown spectroscopy for steel-grade intermix detection in sequence casting

operations. At-line monitoring of temporal evolution versus predicted mathematical model. Spectrochimica Acta Part B: Atomic Spectroscopy. 2018 Aug 1;146:93-100.

[58] Delgado T, Ruiz J, Cabalín LM, Laserna JJ. Distinction strategies based on discriminant function analysis for particular steel grades at elevated temperature using stand-off LIBS. Journal of Analytical Atomic Spectrometry. 2016;31(11):2242-52.

[59] Mier D, Jalali PN, Lopez PR, Gurel J, Strondl A, Cabalín LM, Delgado T, Ruiz J, Laserna J, Campanella B, Legnaioli S. A stochastic model of the process of sequence casting of steel, taking into account imperfect mixing. Applied Physics B. 2019 Apr 1;125(4):65.

[60] Ruiz J, González A, Cabalín LM, Laserna JJ. On-line laser-induced breakdown spectroscopy determination of magnesium coating thickness on electrolytically galvanized steel in motion. Applied spectroscopy. 2010 Dec 1;64(12):1342-9.

[61] Balzer H, Hoehne M, Noll R, Sturm V. New approach to online monitoring of the Al depth profile of the hot-dip galvanised sheet steel using LIBS. Analytical and bioanalytical chemistry. 2006 May 1;385(2):225-33.

[62] Balzer H, Hoehne M, Sturm V, Noll R. Online coating thickness measurement and depth profiling of zinc coated sheet steel by laser-induced breakdown spectroscopy. Spectrochimica Acta Part B: Atomic Spectroscopy. 2005 Aug 31;60(7-8):1172-8.

[63] Nagy TO, Pacher U, Giesriegl A, Weimerskirch MJ, Kautek W. Depth profiling of galvanoaluminium–nickel coatings on steel by UV-and VIS-LIBS. Applied Surface Science. 2017 Oct 1;418:508-16.

[64] Pacher U, Dinu M, Nagy TO, Radvan R, Kautek W. Multiple wavelength stratigraphy by laser-induced breakdown spectroscopy of Ni-Co alloy coatings on steel. Spectrochimica Acta Part B: Atomic Spectroscopy. 2018 Aug 1;146:36-40.

[65] ElFaham MM, Okil M, Mostafa AM. Limit of detection and hardness evaluation of some steel alloys utilizing optical emission spectroscopic techniques. Optics & Laser Technology. 2018 Dec 1;108:634-41.

[66] Afgan MS, Hou Z, Wang Z. Quantitative analysis of common elements in steel using a handheld μ-LIBS instrument. Journal of Analytical Atomic Spectrometry. 2017;32(10):1905-15.

[67] Sorrentino F, Carelli G, Francesconi F, Francesconi M, Marsili P, Cristoforetti G, Legnaioli S, Palleschi V, Tognoni E. Fast analysis of complex metallic alloys by double-pulse time-integrated laser-induced breakdown spectroscopy. Spectrochimica Acta Part B: Atomic Spectroscopy. 2009 Oct 1;64(10):1068-72.

[68] Sturm V, Peter L, Noll R. Steel analysis with laser-induced breakdown spectrometry in the vacuum ultraviolet. Applied Spectroscopy. 2000 Sep 1;54(9):1275-8.

[69] Girón D, Delgado T, Ruiz J, Cabalín LM, Laserna JJ. In-situ monitoring and characterization of airborne solid particles in the hostile environment of a steel industry using stand-off LIBS. Measurement. 2018 Feb 1;115:1-0.

[70] Asimellis G, Hamilton S, Giannoudakos A, Kompitsas M. Controlled inert gas environment for enhanced chlorine and fluorine detection in the visible and near-infrared by laser-induced breakdown spectroscopy. Spectrochimica Acta Part B: Atomic Spectroscopy. 2005 Aug 31;60(7-8):1132-9.

[71] Menneveux J, Wang F, Lu S, Bai X, Motto-Ros V, Gilon N, Chen Y, Yu J. Direct determination of Ti content in sunscreens with laser-induced breakdown spectroscopy: Line selection method for high TiO2 nanoparticle concentration. Spectrochimica Acta Part B: Atomic Spectroscopy. 2015 Jul 1;109:9-15.

[72] Alfarraj BA, Bhatt CR, Yueh FY, Singh JP. Evaluation of optical depths and self-absorption of strontium and aluminum emission lines in laser-induced breakdown spectroscopy (LIBS). Applied spectroscopy. 2017 Apr 1;71(4):640-50.

[73] Ismail MA, Imam H, Elhassan A, Youniss WT, Harith MA. LIBS limit of detection and plasma parameters of some elements in two different metallic matrices. Journal of analytical atomic spectrometry. 2004;19(4):489-94.

[74] Liu X, Ma Q, Liu S, Shi X, Zhang Q, Wu Z, Qiao Y. Monitoring As and Hg variation in An-Gong-Niu-Huang Wan (AGNH) intermediates in a pilot scale blending process using laser-induced breakdown spectroscopy. Spectrochimica Acta Part A: Molecular and Biomolecular Spectroscopy. 2015 Dec 5;151:547-52.

[75] Gondal MA, Habibullah YB, Baig U, Oloore LE. Direct spectral analysis of tea samples using 266 nm UV pulsed laser-induced breakdown spectroscopy and cross validation of LIBS results with ICP-MS. Talanta. 2016 May 15;152:341-52.

[76] Mukherjee D, Cheng MD. Characterization of carbon-containing aerosolized drugs using laser-induced breakdown spectroscopy. Applied spectroscopy. 2008 May 1;62(5):554-62.

[77] Nisar S, Dastgeer G, Shafiq M, Usman M. Qualitative and semi-quantitative analysis of health-care pharmaceutical products using laser-induced breakdown spectroscopy. Journal of pharmaceutical analysis. 2019 Feb 1;9(1):20-4.

[78] Liu X, Ma Q, Liu S, Shi X, Zhang Q, Wu Z, Qiao Y. Monitoring As and Hg variation in An-Gong-Niu-Huang Wan (AGNH) intermediates in a pilot scale blending process using laser-induced breakdown spectroscopy. Spectrochimica Acta Part A: Molecular and Biomolecular Spectroscopy. 2015 Dec 5;151:547-52.

[79] Shen T, Li W, Zhang X, Kong W, Liu F, Wang W, Peng J. High-Sensitivity Determination of Nutrient Elements in Panax notoginseng by Laser-induced Breakdown Spectroscopy and Chemometric Methods. Molecules. 2019 Jan;24(8):1525.

[80] Moncayo S, Duponchel L, Mousavipak N, Panczer G, Trichard F, Bousquet B, Pelascini F, Motto-Ros V. Exploration of megapixel hyperspectral LIBS images using principal component analysis. Journal of Analytical Atomic Spectrometry. 2018;33(2):210-20.

[81] Zou L, Stenslik MJ, Giles MB, Ormes JD, Marsales M, Santos C, Kassim B, Smith JP, Gonzalez JJ, Bu X. Direct visualization of drug release in injectable implant by laser induced breakdown spectroscopy (LIBS). Journal of Analytical Atomic Spectrometry. 2019;34(7):1351-4.

[82] Dubey A, Boukouvala F, Keyvan G, Hsia R, Saranteas K, Brone D, et al. Improvement of Tablet Coating Uniformity Using a Quality by Design Approach. 2012;13(1).

[83] Madamba MC, Mullett WM, Debnath S, Kwong E. Characterization of Tablet Film Coatings Using a Laser-Induced Breakdown Spectroscopic Technique. 2007;8(4):1–7.

[84] Okoyama MY, Ourigny MT, Oroshima KM, Uzuki JS, Akai MS. A Novel Rapid Quantitative Analysis of Drug Migration on Tablets Using Laser Induced Breakdown Spectroscopy. 2010;58(November):2–5.

[85] Myakalwar AK, Dingari NC, Dasari RR, Barman I, Gundawar MK. Non-Gated Laser Induced Breakdown Spectroscopy Provides a Powerful Segmentation Tool on Concomitant Treatment of Characteristic and Continuum Emission. PLoS One. 2014;9(8):e103546.

[86] Myakalwar AK, Sreedhar S, Barman I, Dingari NC, Venugopal Rao S, Prem Kiran P, et al. Laser-induced breakdown spectroscopy-based investigation and classification of pharmaceutical tablets using multivariate chemometric analysis. Talanta. 2011;87(1):53–9.

[87] Yang CS, Jin F, Swaminathan SR, Patel S, Ramer ED, Trivedi SB, Brown EE, Hommerich U, Samuels AC. Comprehensive study of solid pharmaceutical tablets in visible, near infrared (NIR), and longwave infrared (LWIR) spectral regions using a rapid simultaneous ultraviolet/visible/NIR (UVN)+ LWIR laser-induced breakdown spectroscopy linear arrays detection system and a fast acousto-optic tunable filter NIR spectrometer. Optics express. 2017 Oct 30;25(22):26885-97.

[88] Zou L, Kassim B, Smith JP, Ormes JD, Liu Y, Tu Q, Bu X. In situ analytical characterization and chemical imaging of tablet coatings using laser induced breakdown spectroscopy (LIBS). Analyst. 2018;143(20):5000-7.

[89] Zanetti TC, Catelani TA, Pereira-Filho ER, Cabral JS. Laser-induced breakdown spectroscopy as a tool for homogeneity measurements in medicine tablets. Laser Physics. 2020 Feb 3;30(3):035701.

[90] Ahmed N, Liaqat U, Rafique M, Baig MA, Tawfik W. Detection of toxicity in some oral antidiabetic drugs using LIBS and LA-TOF-MS. Microchemical Journal. 2020 Jun 1;155:104679.

[91] Arantes de Carvalho GG, Nunes LC, Florêncio de Souza P, Krug FJ, Alegre TC, Santos Jr. D. Evaluation of laser induced breakdown spectrometry for the determination of macro and micronutrients in pharmaceutical tablets. J Anal At Spectrom. 2010;25(6):803.

[92] Dubey A, Keyvan G, Hsia R, Saranteas K, Brone D, Misra T, et al. Analysis of pharmaceutical tablet coating uniformity by laser-induced breakdown spectroscopy (LIBS). J Pharm Innov. 2011;6(2):77–87.

[93] Eliezer S. The interaction of high-power lasers with plasmas. CRC press; 2002 Aug 16.

[94] Carroll PK, Kennedy ET. Laser-produced plasmas. Contemp Phys. 1981;22(1):61–96.

[95] Griem HR, Lovberg RH. Plasma physics. Academic Press; 1971.

[96] Chen FF. Introduction to plasma physics and controlled fusion. New York: Plenum press; 1984.

[97] Jiang X. Dual-pulse laser induced breakdown spectroscopy in the vacuum ultraviolet with ambient gas: spectroscopic analysis and optimization of limit of detection of carbon and sulfur in steel (Doctoral dissertation, Dublin City University); 2012.

[98] O'Leary EP. VUV laser-induced plasma spectroscopy for low level sulphur detection in steel (Doctoral dissertation, Dublin City University); 2007.

[99] Fallon C. Optical diagnostics of colliding laser produced plasmas: towards next generation plasma light sources (Doctoral dissertation, Dublin City University); 2013.

[100] Hughes TP. Plasmas and laser light. New York, Halsted Press. 1975.

[101] Davitt S. Colliding plasmas in air (Doctoral dissertation, Dublin City University); 2019.

[102] RWP McWhirter.Plasma Diagnostic Techniques. New York: Academic Press; 1965.

[103] Thorne A, U Litzen, S. Johansson. Spectrophysics: Principles and Applications. Springer, 1999.

[104] Colombant D, Tonon GF. X-ray emission in laser-produced plasmas. Journal of Applied Physics. 1973 Aug;44(8):3524-37.

[105] Sizyuk V, Hassanein A, Morozov V, Tolkach V, Sizyuk T, Rice B. Numerical Simulation of Laser-Produced Plasma Devices for EUV Lithography Using the Heights Integrated Model. Numerical Heat Transfer, Part A: Applications. 2006 49:215-236.

[106] Filevich J, Rocca JJ, Jankowska E, Hammarsten EC, Kanizay K, Marconi MC, Moon SJ, Shlyaptsev VN. Two-dimensional effects in laser-created plasmas measured with soft-x-ray laser interferometry. Phys. Rev. E. 2006 67:056409.

[107] Doucet FR, Faustino PJ, Sabsabi M, Lyon RC. Quantitative molecular analysis with molecular bands emission using laser-induced breakdown spectroscopy and chemometrics. Journal of Analytical Atomic Spectrometry. 2008;23(5):694-701.

[108] Tiwari PK, Rai NK, Kumar R, Parigger CG, Rai AK. Atomic and molecular laser-induced breakdown spectroscopy of selected pharmaceuticals. Atoms. 2019 Sep;7(3):71.

[109] Tiwari PK, Awasthi S, Kumar R, Anand RK, Rai PK, Rai AK. Rapid analysis of pharmaceutical drugs using LIBS coupled with multivariate analysis. Lasers in medical science. 2018 Feb 1;33(2):263-70.

[110] MathWorks, Ellipsoid [Internet], Natick, Massachusetts, MathWorks, [Cited 2018 January 17th]. Available from: https://uk.mathworks.com/help/matlab/ref/ellipsoid.html

[111] Wang J, Liao X, Zheng P, Xue S, Peng R. Classification of Chinese herbal medicine by laser-induced breakdown spectroscopy with principal component analysis and artificial neural network. Analytical Letters. 2018 Mar 4;51(4):575-86.

[112] Dingari NC, Barman I, Myakalwar AK, Tewari SP, Kumar Gundawar M. Incorporation of support vector machines in the LIBS toolbox for sensitive and robust classification amidst unexpected sample and system variability. Analytical chemistry. 2012 Mar 20;84(6):2686-94.

[113] Yu J, Hou Z, Sheta S, Dong J, Han W, Lu T, et al. Provenance classification of nephrite jades using multivariate LIBS: A comparative study. Anal Methods. 2018;10(3):281–9.

[114] Zhang C, Shen T, Liu F, He Y. Identification of Coffee Varieties Using Laser-Induced Breakdown Spectroscopy and Chemometrics. Sensors. 2017;18(1):95.

[115] Vahid Dastjerdi M, Mousavi SJ, Soltanolkotabi M, Nezarati Zadeh A. Identification and Sorting of PVC Polymer in Recycling Process by Laser-Induced Breakdown Spectroscopy (LIBS) Combined with Support Vector Machine (SVM) Model. Iran J Sci Technol Trans A Sci. 2018;42(2):959–65.

[116] Li X, Yang S, Fan R, Yu X, Chen D. Discrimination of soft tissues using laser-induced breakdown spectroscopy in combination with k nearest neighbors (kNN) and support vector machine (SVM) classifiers. Opt Laser Technol. 2018;102:233–9.

[117] Lu S, Dong M, Huang J, Li W, Lu J, Li J. Estimation of the aging grade of T91 steel by laser-induced breakdown spectroscopy coupled with support vector machines. Spectrochim Acta - Part B At Spectrosc. 2018;140:35–43.

[118] Tang Y, Guo Y, Sun Q, Tang S, Li J, Guo L, Duan J. Industrial polymers classification using laser-induced breakdown spectroscopy combined with self-organizing maps and K-means algorithm. Optik. 2018 Jul 1;165:179-85.

[119] Costa VC, Castro JP, Andrade DF, Babos DV, Garcia JA, Sperança MA, Catelani TA, Pereira-Filho ER. Laser-induced breakdown spectroscopy (LIBS) applications in the chemical analysis of waste electrical and electronic equipment (WEEE). TrAC Trends in Analytical Chemistry. 2018 Nov 1;108:65-73.

[120] Pagnotta S, Grifoni E, Legnaioli S, Lezzerini M, Lorenzetti G, Palleschi V. Comparison of brass alloys composition by laser-induced breakdown spectroscopy and self-organizing maps. Spectrochim Acta Part B At Spectrosc. 2015;103–104:70–5.

[121] Tonazzini A, Salerno E, Abdel-Salam ZA, Harith MA, Marras L, Botto A, Campanella B, Legnaioli S, Pagnotta S, Poggialini F, Palleschi V. Analytical and mathematical methods for revealing hidden details in ancient manuscripts and paintings: A review. Journal of advanced research. 2019 May 1;17:31-42.

[122] Botto A, Campanella B, Legnaioli S, Lezzerini M, Lorenzetti G, Pagnotta S, Poggialini F, Palleschi V. Applications of laser-induced breakdown spectroscopy in cultural heritage and archaeology: a critical review. Journal of Analytical Atomic Spectrometry. 2019;34(1):81-103.

[123] Raneri S, Pagnotta S, Lezzerini M, Legnaioli S, Palleschi V, Columbu S, Neri NF, Mazzoleni P. Examining the reactivity of volcanic ash in ancient mortars by using a micro-chemical approach. Mediterranean Archaeology & Archaeometry. 2018 Oct 1;18(5).

[124] Arias C, Bani S, Catalli F, Lorenzetti G, Grifoni E, Legnaioli S, Pagnotta S, Palleschi V. X-ray fluorescence analysis and self-organizing maps classification of the Etruscan gold coin collection at the Monetiere of Florence. Applied spectroscopy. 2017 May;71(5):817-22.

[125] Pagnotta S, Lezzerini M, Campanella B, Gallello G, Grifoni E, Legnaioli S, Lorenzetti G, Poggialini F, Raneri S, Safi A, Palleschi V. Fast quantitative elemental mapping of highly inhomogeneous materials by micro-Laser-Induced Breakdown Spectroscopy. Spectrochimica Acta Part B: Atomic Spectroscopy. 2018 Aug 1;146:9-15.

[126] Chengxu LU, Bo WA, Jiang X, Zhang J, Kang NI, Yanwei YU. Detection of K in soil using time-resolved laser-induced breakdown spectroscopy based on convolutional neural networks. Plasma Science and Technology. 2018 Dec 27;21(3):034014.

[127] Chen J, Pisonero J, Chen S, Wang X, Fan Q, Duan Y. Convolutional neural network as a novel classification approach for laser-induced breakdown spectroscopy applications in lithological recognition. Spectrochimica Acta Part B: Atomic Spectroscopy. 2020 Apr 1;166:105801.

2 Experimental apparatus and methods

In essence laser induced breakdown spectroscopy (LIBS) is a very simple analytical technique. It requires a laser, a system to deliver the laser pulse to the sample under analysis to form a plasma, a method to collect the light emitted from the plasma and a spectrometer to record this emission spectrum. When moving to the vacuum ultraviolet (VUV) region this set-up needs to be modified somewhat. The target and detection systems need to be under vacuum in order to prevent the strong absorption of VUV in air, and most gases. Traditional collection optics, such as fibers, lens systems, etc., are also not possible due to the lack of refractive optics in the VUV spectral region. To overcome this the emission from the plasma passes directly into the entrance slit of the spectrometer. Compact spectrometers used for LIBS in the visible region are also not possible in the VUV, in this case a normal incidence, off-Rowland circle mounted grating spectrometer is used. The use of VUV-LIBS light also has implications for the detector – intensified charge-coupled devices, which are standard in modern LIBS systems are again not possible, so a back-illuminated CCD camera is used instead to provide sensitivity in the VUV.

A schematic of the final system used is given in figure 2.1, with the major components labeled. In section 2.1 details are given for each of these components in turn, as well as details on the steps required to synchronize the laser pulse to the detector.

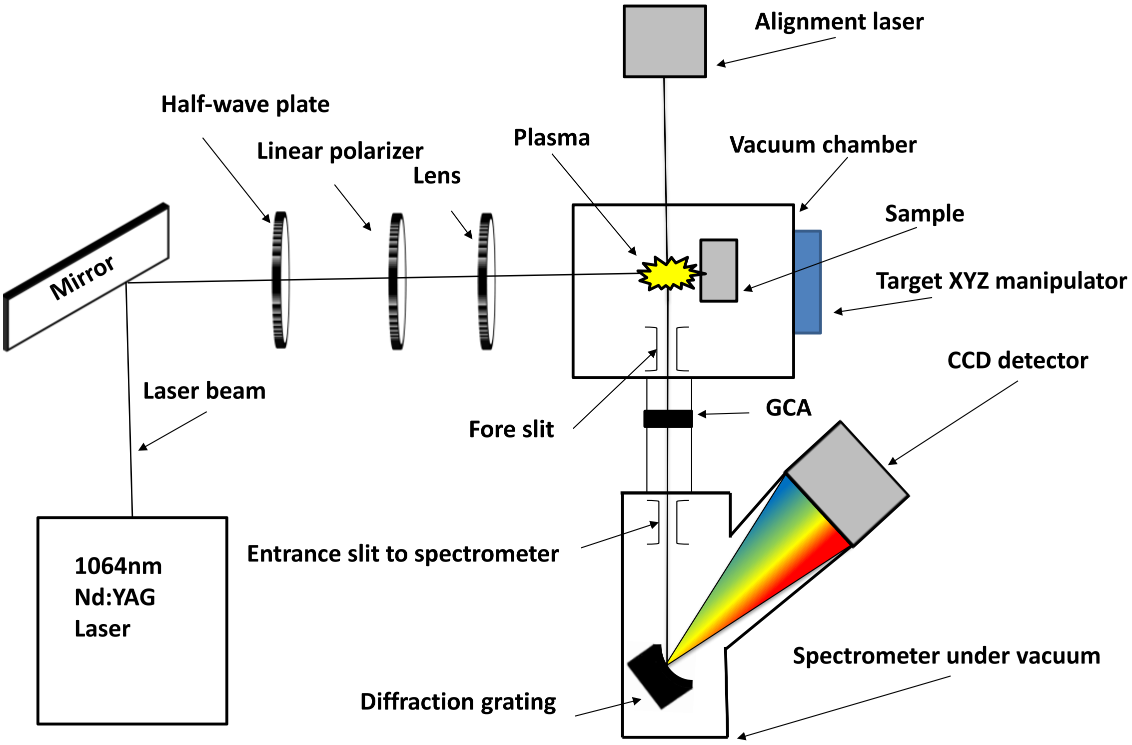


Figure 2.1: Schematic, indicating the main components, of the experimental set-up used for the VUV-LIBS studies present in this thesis.

The main components of system in DCU were provided – historically the system was used for LIBS analysis on steel targets [1-5]. My contributions were introducing a system to control the pulse energy onto the targets, designing and incorporating the target mount for pharmaceutical tablets, performing the required optimisation to probe the tablets and acquiring all the spectra.

The change from steel to a pharmaceutical tablet is significant. The tablets are grainy non-cohesive structures consisting of loosely compacted powder require a different method to mount the target samples. The lack of cohesion leads to the observation of macroscale fragment ejection, and the requirement to have better control of the laser pulse energy. A system which provides reproducible spectra, across a diverse sample type, is needed to provide robust analysis. Initial investigations of these samples identified some substantial difficulties in providing spectra for this robust analysis. These include the inhomogeneous nature of the samples, significant height variations across a sample, varying sample sizes and delamination of loosely compacted samples under laser pulse irradiation. The pharmaceutical tablets had curved edges and probing on curved edges led to observed inconsistencies in the spectra, which are not necessarily due to differences in composition. It was necessary to probe the flattest section of the tables to record reliable spectra. Vernier calipers were used to measure the thickness of the samples. When that thickness tapered off it was noted and only the flat face was probed. Procedural work was conducted to find the optimal pulse energy, the distance from the optical axis to view the plasma from and the inter shot spacing required to avoid probing melt zones. Samples were probed from the face edge furthest away from the slit then moving towards the slit so ridges from melt zones and craters from previous plasmas would not obscure emission from subsequent plasmas. Section 2.2 details the experimental parameters used. An outline for the initial parameters is shown in section 2.2.1 and the justification for the optimisations are made in section 2.2.2. Finally, once the set-up and optimization of the experimental apparatus was complete a protocol for recording spectra across the different sample type was developed. Details of the pharmaceutical tablets used in this thesis are shown in section 2.3.

2.1 Experimental apparatus

The basis of the VUV-LIBS system has been developed in DCU for a number of years and has proven to be adaptable to a number of scientific investigations [1-5]. The final configuration used in this study is presented in figure 2.1. In this section details are provided on the individual components which make up this VUV-LIBS system.

2.1.1 Laser system

At the heart of any LIBS system is the drive laser. In this case a Continuum Surelite III-10 laser was employed [6]. This provided a maximum pulse energy of 460 mJ at a wavelength of 1064 nm and a pulse duration of 6 ns. This is more than sufficient to generate plasmas which emit in the VUV spectral region. The laser is a Nd:YAG system. A schematic of the internal optical layout of the Surelite laser is shown in figure 2.2. The lasing process begins with the laser head (component 1). This consists of a xenon flashlamp and the 9 mm diameter, 115 mm long Nd:YAG crystal laser rod, with antireflection coatings at either end. The xenon flashlamp produces broadband radiation extending from the UV to IR spectral regions and is used to pump the lasing system.

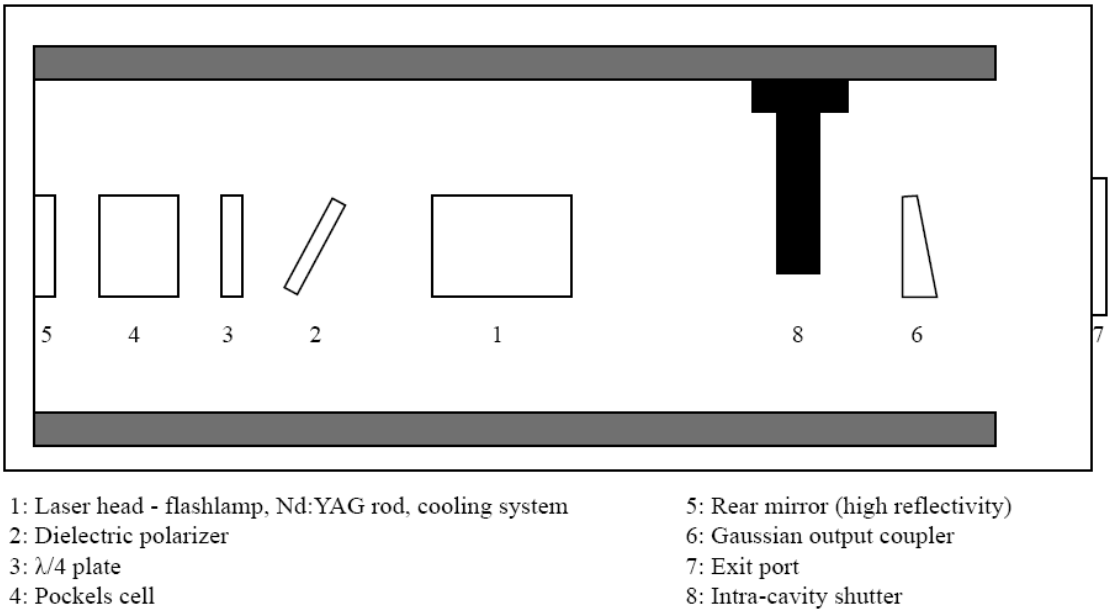


Figure 2.2: Schematic diagram of the optical layout of the Surelite III-10.

The combination of the dielectric polarizer (component 2), λ/4 plate (component 3) and Pockels cell (component 4) is referred to as the Q-switch. The principle of Q-switching is to keep the gain of the resonator low until maximum population inversion is achieved [7-16]. Once this occurs the quality (Q) factor is decreased, allowing lasing to take place. A Pockels cell is an optically active crystal, where changing the applied voltage leads to a change in the birefringence of the crystal. As such it behaves as a voltage dependent wave plate. If no voltage is applied to the Pockels it will have no impact on the polarization of the light passing through it. However, at the quarter wave or '$\lambda/4$' voltage a phase a shift of 90 degrees is introduced between the ordinary and extraordinary waves passing through the cell. In this case an initially linearly polarized beam is converted to a circularly polarized beam.

In the Surelite, the horizontal component of light leaving the rod is allowed to remain in the cavity travelling towards the rear mirror (component 5), while the vertical component is rejected. As the beam propagates within the cavity it makes a double pass through the Pockels cell and $\lambda/4$ plate. With no bias on the Pockels cell the cavity remains closed, the original horizontally polarized beam is rotated through 90 degrees by the double pass through the $\lambda/4$ plate and is therefore rejected by the polarizer.

When a high ($\lambda/4$) voltage (~ 3 kV) is applied to the birefringent Pockels cell crystal, the laser field undergoes a double pass through the Pockels cell and a double pass through the $\lambda/4$ plate so that the beam polarization is returned to its original horizontal state and hence it is no longer rejected by the polarizer and is allowed pass through the laser rod and experience significant gain.

As this process can be controlled by the application of the voltage to the Pockels cell it is considered an active Q-switch. The advantages of active Q-switches are the Q value can be changed very quickly to give the 6 ns optical pulse, at the moment of optimum population inversion, in this case 235.8 μs after triggering the flashlamps.

However, the laser light produced in the Surelite is still located within the cavity and has a poor spatial and angular distribution. The laser light is allowed to exit the cavity and is partially improved using the Gaussian output coupler (component 6). This is a mirror with variable transmission across its surface, high transmission at the centre and low at the edges. The laser pulse is then free to pass through the exit port (component 7) for use in experiments. The intra cavity shutter (component 8) allows the production of laser light to be prevented at any stage and is both manually controlled and connected to the safety interlock system.

2.1.2 Free-standing optical components

Once the laser beam passes the exit port it must be steered, energy controlled and focused onto the samples of interest. This is achieved using a set of free-standing optical components. The first of these is a periscope, constructed using two dielectric mirrors, with a coating to ensure high reflectivity at 1064 nm. This periscope arrangement is used to adjust the height and direction of the laser beam so that it is incident close to the sample normal. A slight misalignment is intentionally included so that no reflected beams can reenter the laser system, which may cause damage to the rod and internal optics of the laser.

Next the laser beam is incident upon a half wave and polarizer, which act in tandem to control the energy of the laser beam. The half wave plate is made of quartz, which displays birefringence. Light which is polarized in the direction of the optic axis of the crystal (extraordinary light) experiences one refractive index, while light polarized perpendicular to the optic axis (ordinary light) experiences a lower refractive index. This means that the extraordinary light will be delayed, relative to the ordinary light, as it passes through the crystal. If the thickness of the crystal is chosen correctly this delay will correspond to half of the wavelength of the light – giving a half wave plate. If linearly polarized light is incident onto the half wave plate the net result is linearly polarized light that is rotated by twice the angle between the polarization of the light and the optic axis of the crystal.

For example, if the light is polarized parallel or perpendicular to the optic axis no change occurs, however if the light is polarized at 45° to the optic axis then vertically polarized light will become linearly polarized light, and *vice versa*. If an analyzer polarizer is then used, then the energy passing through both can be finely controlled. If the half wave plate rotates the laser light such than it is aligned with the transmission axis of the polarizer then all of the light will pass through. However, if the half wave plate is positioned so that it rotates the laser light such that it is polarized perpendicular to the transmission axis then no light will pass. The positioning of the half wave plate is achieved by a simple rotation. The laser pulse energy I recorded after the polarizer as a function of half wave plate angle is shown in figure 2.3

The final two optical components in the path of the laser beam are the lens and the entrance window to the target chamber. The lens used in these experiments is a 100 mm focal plano-convex lens with a 1064 nm antireflection coating. This was positioned on a translation stage that could be moved in the x, y, and z direction. The x-y motion was utilized to ensure the laser beam passed through the centre of the lens, while the z motion controls the distance from the lens to the surface of the sample. This controls the focal spot size, the closer to focus the smaller the spot size, and can be moved in conjunction with the target to ensure a fixed spot size on the sample.

Finally, the laser beam is allowed to enter the target chamber via a large entrance window, which has a recorded transmission of 80%.

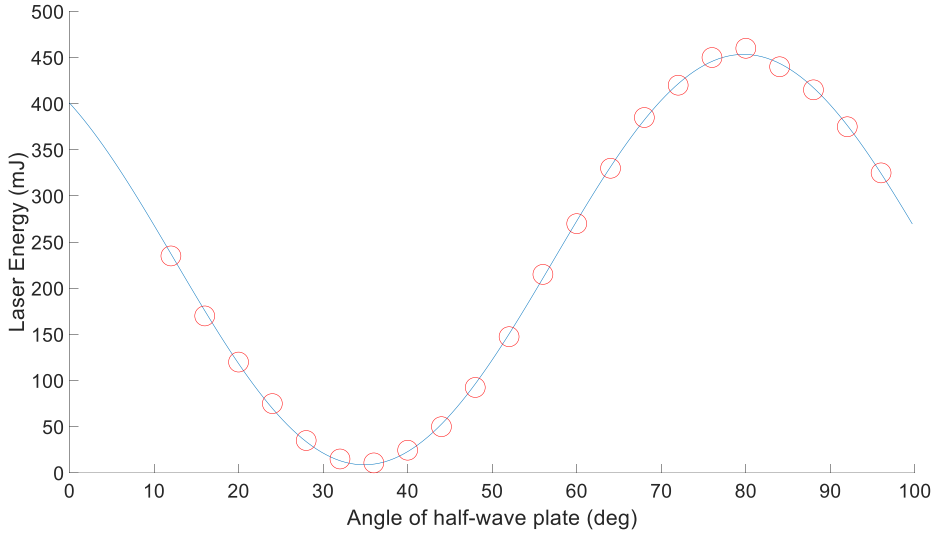


Figure 2.3: Recorded laser energy vs half-wave plate angle

2.1.3 Target chamber and holder

The target chamber used in the experimental setup is an aluminum cube. The cube has sides of 12 cm and the interior is hollow, having been drilled from face to face with a cylinder radius of 3.5 cm to give a total internal volume of 0.75 liters. The sides were polished to fit O-ring sealed flanges and a rubber O-ring is used to form a vacuum seal between the window, and all other ports and feedthroughs, and the chamber. The target chamber is evacuated by a turbomolecular pump backed with a double-stage rotary pump to a pressure of ~2 x 10^{-6} mbar.

As this operating pressure was maintained across all studies, the time taken to acquire spectra was the same for each sample. It took approximately 30 minutes from placing a sample in the target holder to acquiring spectra. It should be noted that this long wait is largely due to the vacuum equipment used and the large volume of the target chamber compared to the relative size of the pharmaceutical tablets. As a pressure difference is maintained between the low pressures of the spectrometer and the chamber, a smaller chamber, e.g., 3x3x3 cm , would be ideal for quick pumping with larger pumps, dramatically reducing the time require to obtain spectra. This would make the process more apllication friendly, for at-line measurements for example. The laser operated at 10 Hz leading to the capability of recording 10 spectra per second, which the camera used can support in kinetic series mode (an exposure time of 0.1 s ensures that emission from the entire lifetime of the plasma is recorded). This rate was not achieved in practice as the target was manually manipulate, which required walking from the laser/camera control station to the target chamber to present a fresh part of the table to the laser pulse when required. However, automated control motors could ideally be used for target manipulation to obtain a laser limited throughput for spectra. Finally, it took approximately 15 minutes to bring the chamber back up to atmospheric pressure, which could be improved using a smaller chamber and/or a venting gas. An idealised system would be limited to approximated 2 s + pumping/venting time to recorded sufficient spectra to exploit the techniques presented in this thesis.

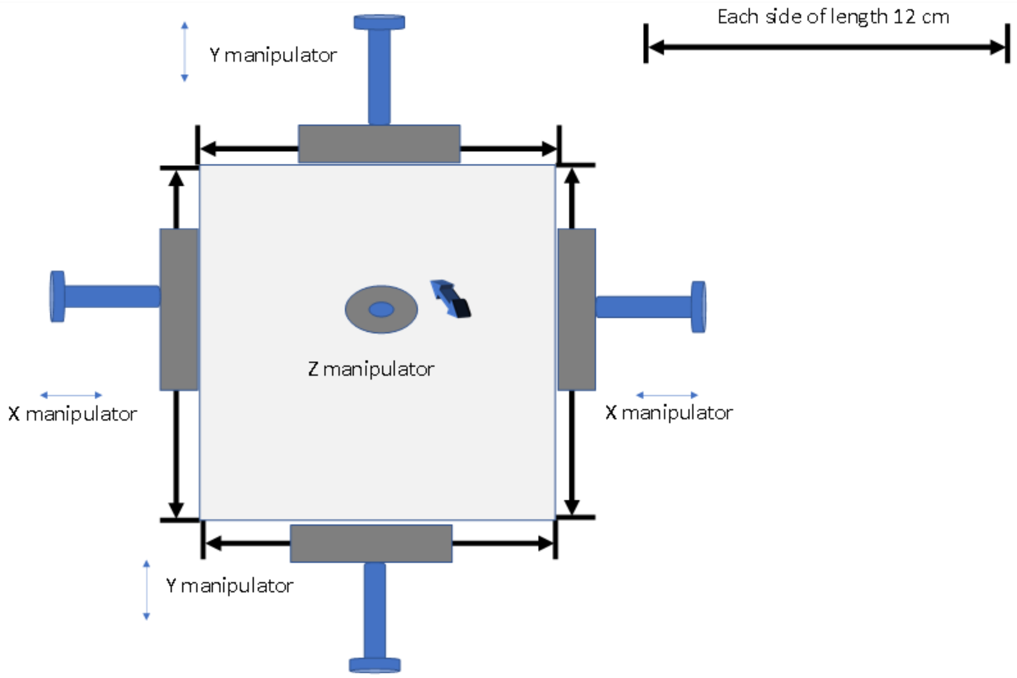


Figure 2.4: Diagram of the external XYZ manipulator to control of the target, locate under vacuum within the chamber.

The target holder is placed inside this chamber. This holder is coupled to an XYZ manipulator by a post, onto which the holder could be screwed. The post can be adjusted by micrometer translational screws to control position in the X, Y and Z directions. The X direction is defined as the horizontal position of the target holder in the chamber. The Y direction is defined as the vertical position of the target holder in the chamber. The Z direction is defined as movement towards/ away from the optical axis of the spectrometer. This allows for 3-dimensional position control of the target holder in the target chamber, as I have drawn schematically in figure 2.4.

The target holder I designed and used in these studies, as seen in figure 2.5, was primitive, but effective. The main body was constructed from aluminium, with five holes. One of these was bored completely through and is used to screw the holder on to the post controlled by the XYZ manipulator. The other four holes are tapped, so that the restraining aluminium bars can secured to the body of the holder. These bars are used to hold the samples in place. Once the sample is attached to the holder a Vernier calipers is used to measure the distance from the rear of the holder to the sample surface. This is then added or subtracted from the Z position of the manipulator to ensure that each sample is positioned at the correct distance from the optical axis of the system. The sample holder is then attached to the post, the system is sealed and finally pumped down to the required pressure.

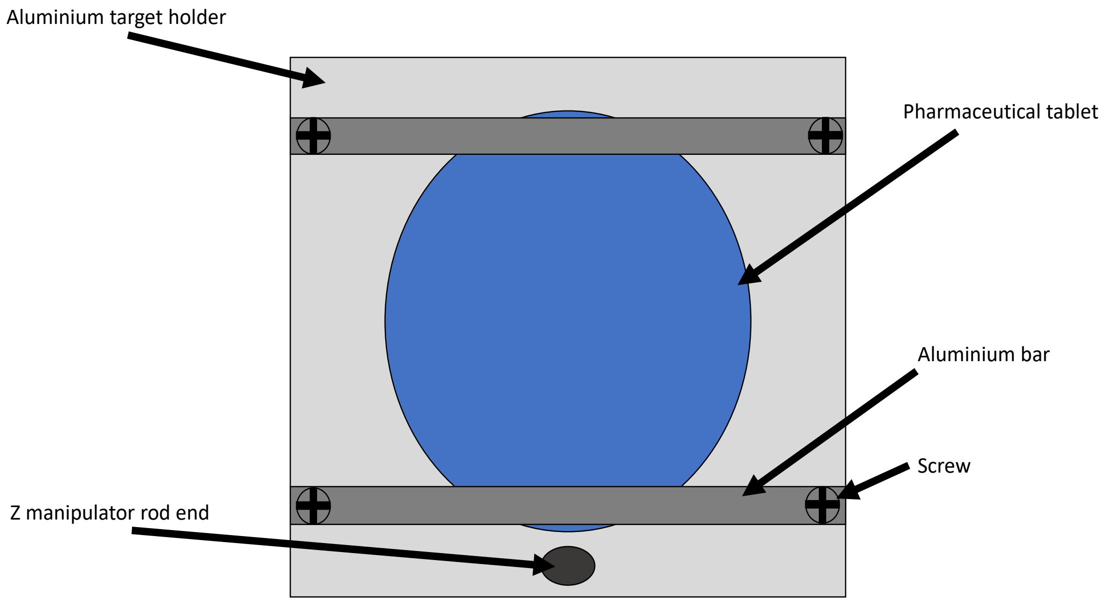


Figure 2.5: Diagram of the target holder used to hold and manipulate the samples under investigation.

2.1.4 Fore-slit and glass capillary array

Also located inside the target chamber is a fore-slit. This is used to block out all but a selected portion of the plasma along the optical axis, to enable spatially resolved LIBS. This slit, of width 250 μm, is position 40 mm from the plasma. When combined with the 50 μm entrance slit to the spectrometer about 370 mm away, this limits the spatial extend of light which can be detected. This results in a spatial resolution of approximately 0.3 mm about the optical axis.

The glass capillary array (GCA) is a component that serves two purposes. Firstly, it allows for pressure differences to be maintained between the target chamber and the spectrometer while allowing light to pass through. Secondly it prolongs the lifetime of the spectrometer by significantly reducing the debris that can enter the spectrometer and potentially coat the grating, leading to a loss in efficiency.

An example of a GCA, by Burle Industries Inc, is shown in figure 2.6. This image shows the general structure of GCAs – many micrometer scale pores through a ceramic, such that it is mainly open.

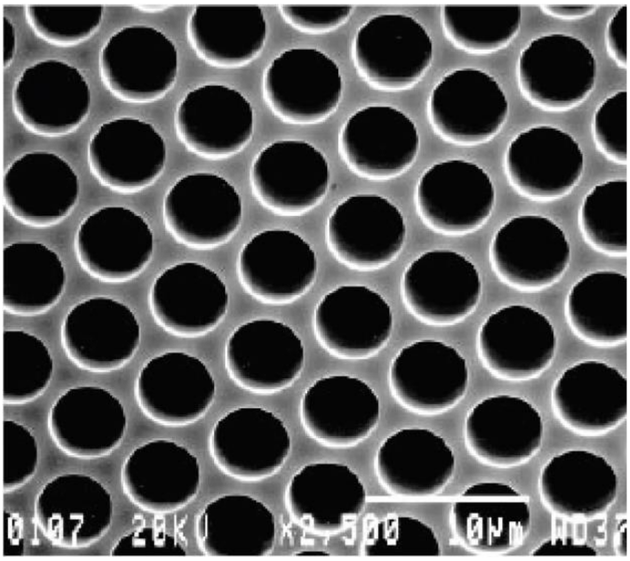


Figure 2.6: Microscope image of a typical GCA, the white scale bar corresponds to 10 μm [17].

The particular GCA used in this study was provided by Collimated Holes Inc. [18] composed of a lead alkali silicate. It measures 25 mm in diameter and is 3 mm thick, with a pore diameter of 50 μm. The advantageous properties of GCAs stem from the small relatively long holes, which results in a high aspect ratio for the pores. This means that as gas molecules pass through the pores there are a large number of collisions, so few gas molecules propagate down the entire length of the pores. However, as a large percentage of the GCA is open, there is little attenuation of the light which passes through it.

2.1.5 Spectrometer

The experiments in the VUV spectral range presented in this thesis are performed using a 1m normal incidence Action Research Corporation VM521 system [1-4, 19] which is equipped with an Al+MgF_2 coated 1200 groves/mm holographically ruled diffraction grating supplied by Bausch and Lomb. The spectral range investigated covered a substantial portion of the VUV, from 37 nm to 112 nm, where the Al/MgF_2 coating on the grating provides enhanced reflection efficiency. A photograph of the spectrometer used is shown in figure 2.7, while a schematic of the spectrometer, in the 0th order position, is also shown in figure 2.8.

The spectrometer consists of essentially only three parts, a 50 μm entrance slit, a concave grating and an exit port where the detector is placed. If all three components are properly located on a circle with a diameter that is equal to the radius of curvature of the grating, i.e., the Rowland circle [20-24], the entrance slit will be imaged, by the grating, onto the exit slit position.

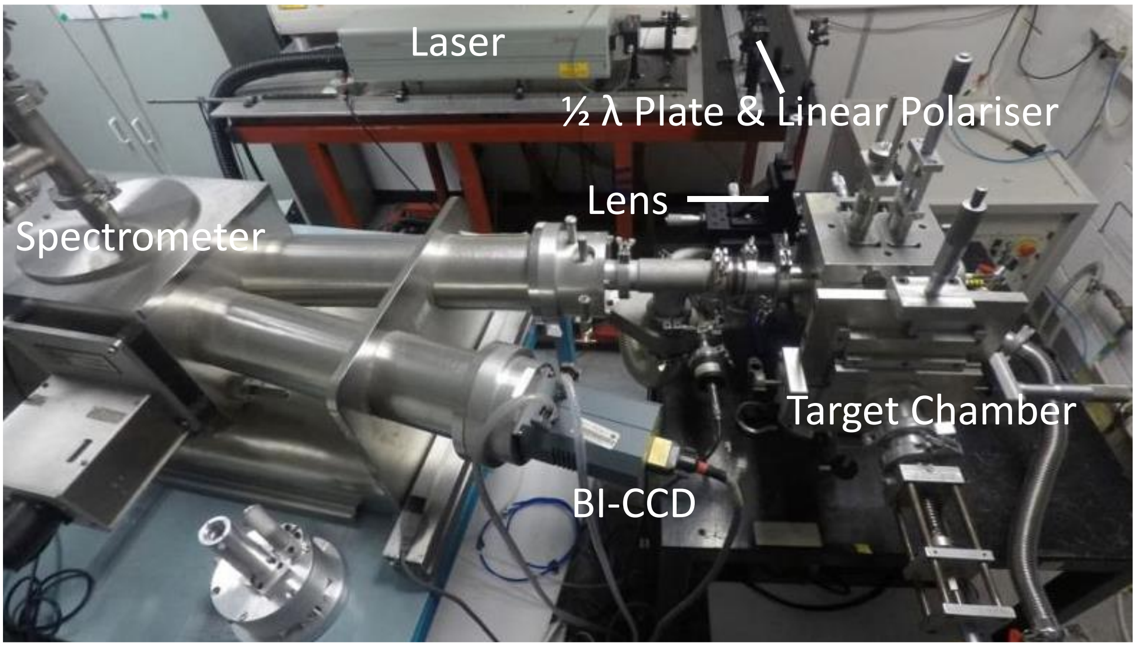


Figure 2.7: View of the VUV spectrometer at DCU.

In the 0th order position, where the angle of incidence and diffraction are equal, specular reflection will result in the formation of a “white light” image known as the central image or zeroth order image. However, given that the entrance and exit arms of the spectrometer are fixed, to record a spectrum of width ~16 nm at a different wavelength region on the detector the diffraction grating must be rotated about an axis parallel to its vertical grooves, and tangential to its centre. The appropriate focussing conditions are maintained at the detector and the grating by linearly translating the grating along the bisector of the angle subtended, 15° in the present instrument, by the slits at the grating centre. It is this rotation and translation of the grating which defines a concave grating use in an off-Rowland circle mount.

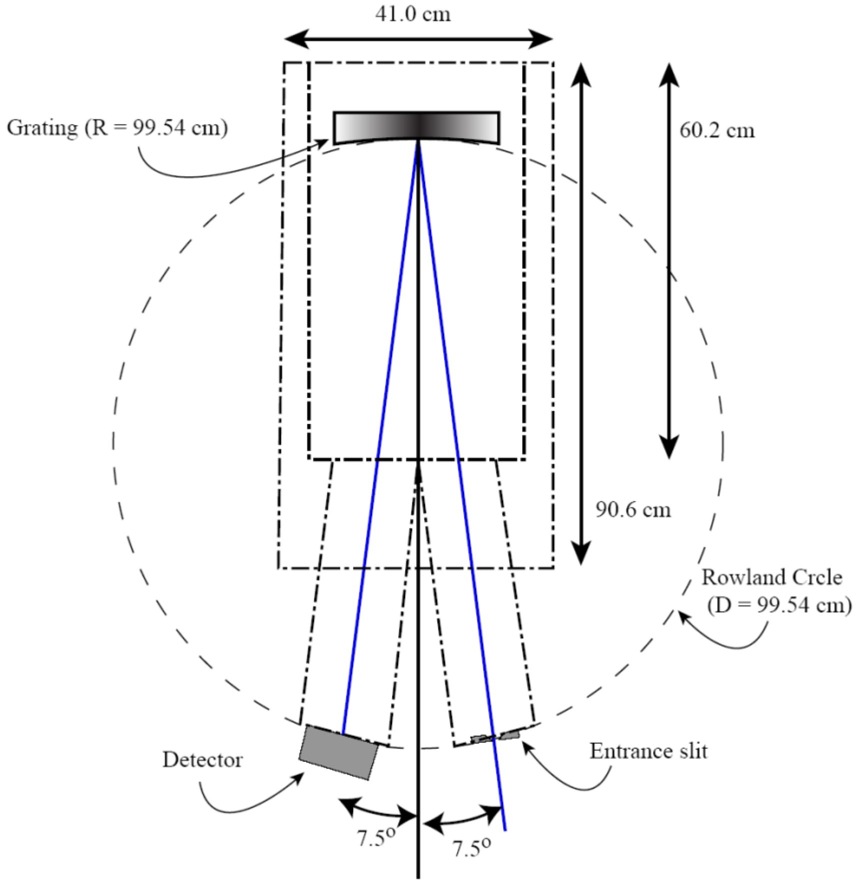


Figure 2.8: Schematic of the VUV spectrometer at DCU.

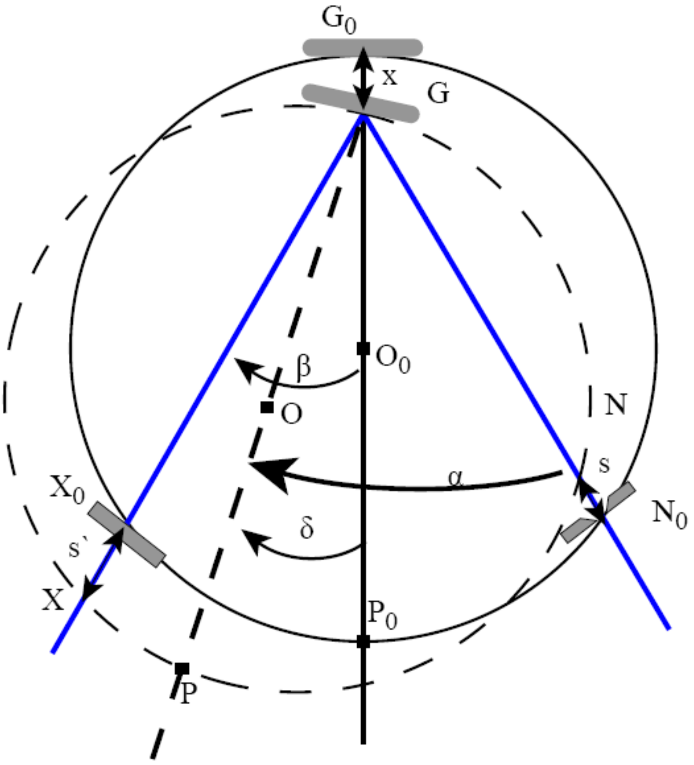


Figure 2.9: Operation of an off-Rowland circle mount spectrometer (see text for details).

Looking at figure 2.9 the terms G_0, N_0, X_0, O_0 and P_0 are the grating, entrance slit, detector, centre of the Rowland circle and the point of intersection of the grating normal with the Rowland circle, respectively, in the initial position. The non-subscript labels refer to the same, but after the rotation through an angle δ and a linear motion through a distance x.

A detailed description of the mathematics involved in an off-Rowland circle mount normal incidence spectrometer is described in [24]. As such only a brief hightlight of some important features will be given here. If the initial position of the spectrometer is taken as that shown in figure 2.8, then by rotating the grating both the entrance slit and the detector will no longer lie on the circumference of the Rowland circle, which will also undergone a rotation, as shown in figure 2.9. The entrance slit is now located a distance, s, beyond the newly rotated Rowland circle, and the centre of the detector plane is located a distance, s'. A detailed analysis [24] of figure 2.9 gives the required linear motion for optimum focussing as:

$$x = \frac{R}{2}\sin^2\delta - \frac{R}{2}\tan^2\alpha_0 \qquad (2.1),$$

which, for small δ can be approximated by

$$x = R(1 - \cos\delta) \qquad (2.2),$$

where α_0 is the angle of incidence before rotation of the grating, R is the diameter of the Rowland circle and δ is the angle through which the grating is rotated.

From the manufacturers manual, to scan a wavelength range from 30 to 325 nm the grating must be rotated through an angle ranging from 0 to 20°, with a corresponding linear translation from 0 to 20 mm. A cam driven stage, machined to give the correct combination of both movements (rotational and linear) for each wavelength setting is used to accurately position the grating. The cam is driven via a precision lead screw located outside the vacuum spectrometer, which is connected to a stepper motor operated by a SpectraDrive™ controller [19]. A single revolution of the stepper motor coressponds to a changes in the central wavelength on the detector of approximately 1 nm. The spectrometer is evacuated, through a flange attached to its base, by a turbomolecular pump backed with a double-stage rotary pump to a pressure of ~2 x 10^{-6} mbar.

2.1.6 Detector

The detector in this study is a back illuminated VUV-sensitive 2-D Andor™ back-thinned CCD camera (or BI-CCD), model number DV420-BN [25], and is used to convert the VUV photons to electronic charges to be quantified. The CCD is comprised of an array of metal oxide semiconductors (MOS) capacitors, each representing a pixel on the CCD (total array size 1024 pixels x 512 pixels). A traditional layout of a MOS capacitor is shown in figure 2.10(a). By applying a positive voltage to the gate (poly-Si) of the sensor, a potential well is formed below the gate contact, this is also called the depletion region since it is free from majority carriers. If light then passes into the capacitor and penetrates into the depletion region of the sensor, one or more electron-hole pairs are produced. These positive or negative charges can then be read and interpreted as light intensity. It is this process that inhibits the use of traditional CCDs in the VUV region: to produce the charges the light must penetrate into the depletion region, but to do so it must pass through the gate and SiO_2 layers, where absorption is very high at short wavelengths.

To overcome this BI-CCDs are used in the VUV. The issues are overcome in two ways, shown in figure 2.10(b). Firstly, the pixel is turned around so that the light enters through the p-doped Si. Secondly, the p-dope Si layer is thinned to a depth of 10–15 μm so that the light must pass through less material to reach the depletion region. Both of these modifications overcome the absorption problems for VUV light in traditional CCD designs. However, a drawback is that the level of dark current is increased, and also leaving the chips more prone to damage.

Once charges are produced, they must then be recorded and sent to a computer. To achieve this the gate voltage is alternated on each pixel, also known as clocking. The charge is transferred down into the shift register, as shown in figure 2.11. The shift register passes the electronic pulses onto an amplifier connected to a 16-bit digitizer card in the PC. Once on screen the output can be viewed, and recorded, as an image or spectrum.

In this study the camera was used in full vertical binning mode. This means the outputs are the sum of vertical columns of pixels. This mode of operation has two advantages: firstly, it reduces the noise present in the spectra as the noise on each pixel averages to zero in the summation; secondly, it can increase the readout speed.

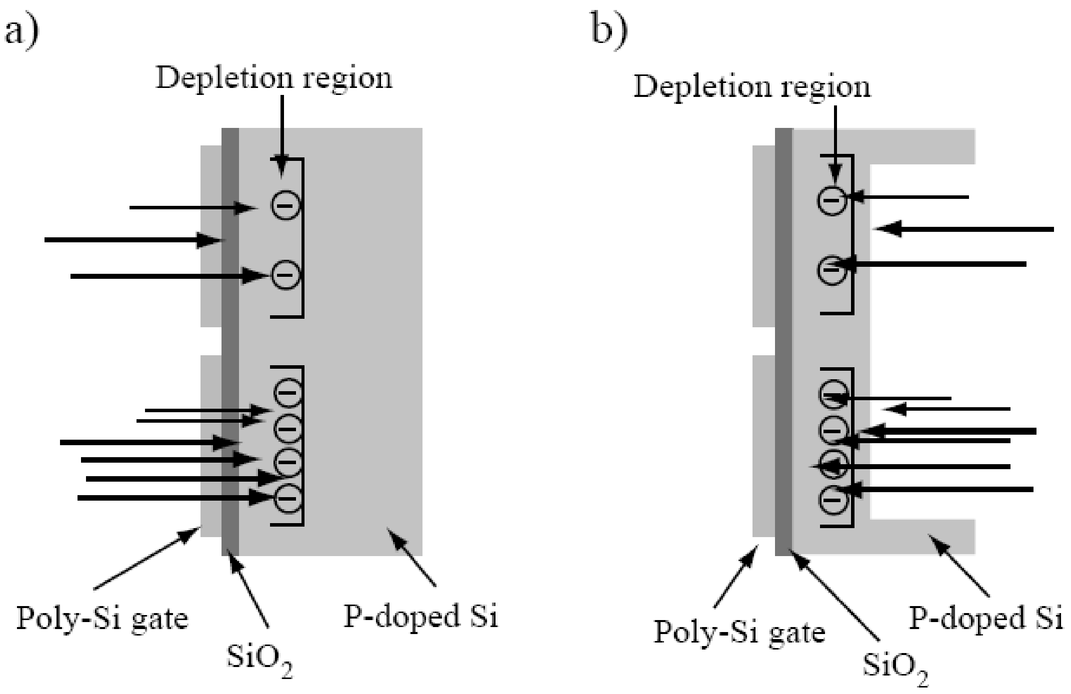


Figure 2.10: a) Traditional CCD structure and b) BI-CCD structure.

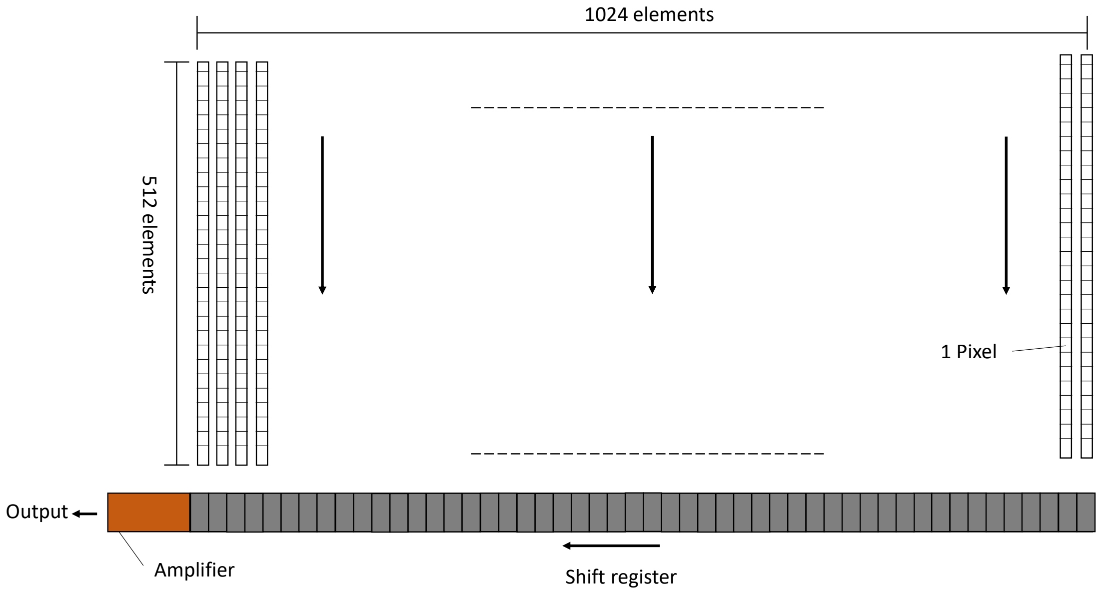


Figure 2.11: MOS capacitor structure.

Water cooling, controlled by the camera software, is used to lower the temperature of the chip to -40°C to reduce thermally generated background signals. Additional noise can be reduced in spectra via software background subtraction, whereby a background spectrum is taken under the same conditions as regular spectra, but without the laser in operation. This background spectrum is subtracted from subsequently taken spectra.

2.1.7 Synchronization of laser and camera

To ensure that the same number of plasmas contributed to each recorded spectrum it is required that the camera and laser are synchronized. To enable this the software to control the camera is supplied with an I/O box, one port of which is set high during the exposure of the chip, and the laser provides direct access to trigger the flashlamps and Q-switch independently. The synchronization was achieve using these outputs and inputs in conjunction with a Quantum Composer Plus 9514 delay generator [26] and an AND gate.

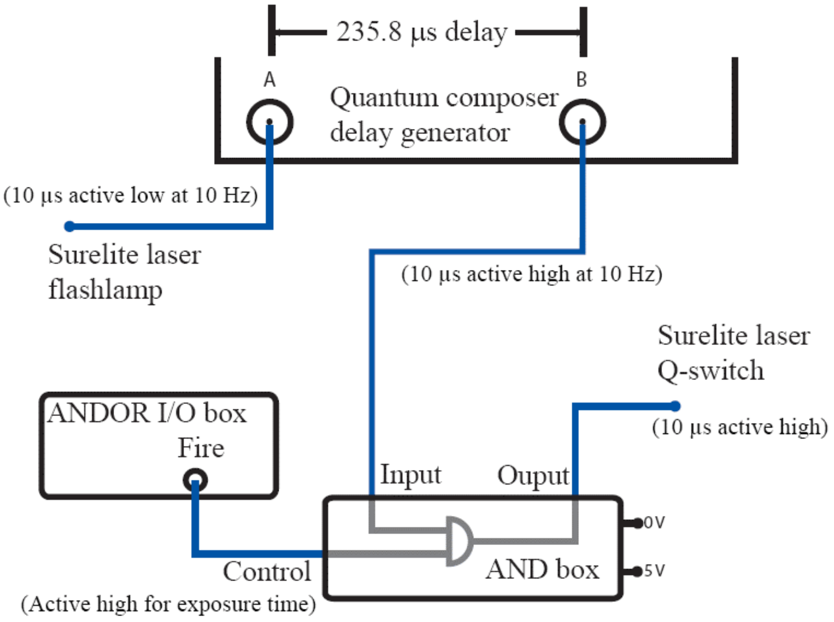


Figure 2.12: Diagram showing the camera-laser synchronization.

The Quantum Composer is set as the master, with two outputs. The first of these is an active low transistor-transistor logic (TTL) signal running at 10 Hz with a 10 μs pulse width which drives the laser flashlamps directly. The continuous triggering of the flashlamps at 10 Hz is important as it maintains thermal stability within the laser cavity. When the “acquire” button is pressed in the software controlling the camera sends it a signal to the output for the duration of the exposure. This output is sent to the AND gate, where the second AND input contains an active high TTL signal running at 10 Hz from the second output on the Quantum Composer. This has a 10 µs pulse width, set-up with a delay of 235.8 µs relative to the flashlamp signal. When both the camera and Q-switch signals are present the 10 µs pulse is passed to the laser activating the Q-switch, allowing the laser to output a 6 ns optical pulse. This scheme is shown in figure 2.12.

The camera was operated in two different ways. In single shot mode the exposure time was set to 0.101 s. This ensures that one and only one laser pulse formed a plasma during the exposure. The second way was in a kinetic series. The exposure was again set to 0.101 seconds, but a series of exposures were recorded with 0.101 seconds between the end of one and the start of the next, as a series of frames. This allowed 2 shots to be taken on one target position in quick succession, but the spectrum for each stored. These could then be investigated individually or summed.

2.2 Experimental conditions and optimization

The first experimental details are described in section 2.2.1, these conditions were used for the initial studies conducted in chapter 4. The steps taken to optimize the set-up following the learnings of chapter 4 are shown in section 2.2.2. These revised experimental conditions were used in chapter 5 for the majority of the studies conducted.

2.2.1 Initial experimental conditions

Initial alignment was performed using a HeNe alignment laser which passed through the fore-slit and entrance slit of the spectrometer. The target was positioned so that the alignment laser grazed along its surface. The lens was positioned so that the laser beam was focused to a spot size of approximately 200 µm diameter. The large spot size diameter was chosen to ensure a large sampling site was taken to account for the sample inhomogeneity. Once this initial position was determined the chamber was evacuated and the spectra could be recorded.

As noted in section 2.1.4 the fore-slit reduced the field of view of the spectrometer to ~ 300 µm, so this could be used to further define the position of the optical axis. The target and lens were moved in tandem, to ensure the spot size remained constant, until the most forward position was found where a spectrum could be obtained while using the full 460 mJ of available laser energy. This was taken as the zero position of the optical axis. The target was then positioned such that it was 2 mm from the optical axis, and a plasma formed on the surface would expand towards the optical axis. It is often desirable to allow the plasma to expand and cool before collecting the radiation from it, to reduce self-absorption and continuum emission, and this 2 mm value was taken from previous set-ups used for quantitative LIBS [4, 5] and agreed with observations recorded, spectra as seen in appendix E. Self-absorption is the absorption of a photon produced in the plasmas by material (atoms/ions) as the light passes through the plasma. Transitions are element specific, quantized and of a specific wavelength, so a given species has the highest probability of reabsorbing the photon emitted by the same species. While self-absorption can be a significant issue when LIBS is used for limit-of-detection purposes (as the linear relationship between line intensity and concentration is no longer valid), it is not expected to be such an issue in the current studies. This work treats the whole spectrum as a single identity for classification purposes, rather than focusing on a single or a few lines – any self-absorption present in the spectra is a feature of the spectra that could be useful for classification purposes.

2.2.2 Revised experimental conditions

Following from the studies conducted in chapter 4, a number of issues were identified, such as the inhomogeneous nature of samples, significant height variations across samples, highly curved surfaces, varying sample sizes and delamination and fragmentation of loosely compacted samples under laser pulse irradiation. To address these issues significant re-alignment and optimization of the experimental system was required in order to: optimize the focusing conditions of the laser pulse onto the sample surface to prevent excessive damage of the surface; determining the optimal spatial separation between sampling sites to prevent overlap of craters created by the laser produced plasma formation; and to find stable positions of the samples, relative to the optical axis of the spectrometer, from which to record the spectra.

Figure 2.13 shows the consequences of using too high a laser energy when probing a target, particularly pharmaceutical tablets that are grainy non-cohesive structures that are designed to break apart easily. In order to avoid blowing large macro-scale ejecta from the samples the laser energy was decreased from 460 mJ to 265 mJ (this corresponded to a half-wave plate setting of 60 degrees). A number of different spot sizes were also investigated, it was chosen to maintain the 200 μm spot size as it accounted well for the sample inhomogeneity.

Another important factor is the spacing out of the formation points of the laser produced plasmas, as shown in figure 2.14. If the plasmas are formed too close together the melt zones at the plasma formation sites can overlap. The melt zone is a heated portion of the sample around the crater and its chemical composition is not indicative of the original sample. Therefore, spacing shots far enough apart is necessary to not sample these melt zones and avoid recording spectra which are not indicative of the samples under investigation. At 0.5 mm spacing there was overlap of melt zones, which continued up to 0.8 mm spacings. However, at 0.9 mm spacings and above there was no overlap observed. Therefore a 1 mm spacing between shots was used, as there was no probing of melt zones whilst still being able to probe enough of the surface of the tablet.

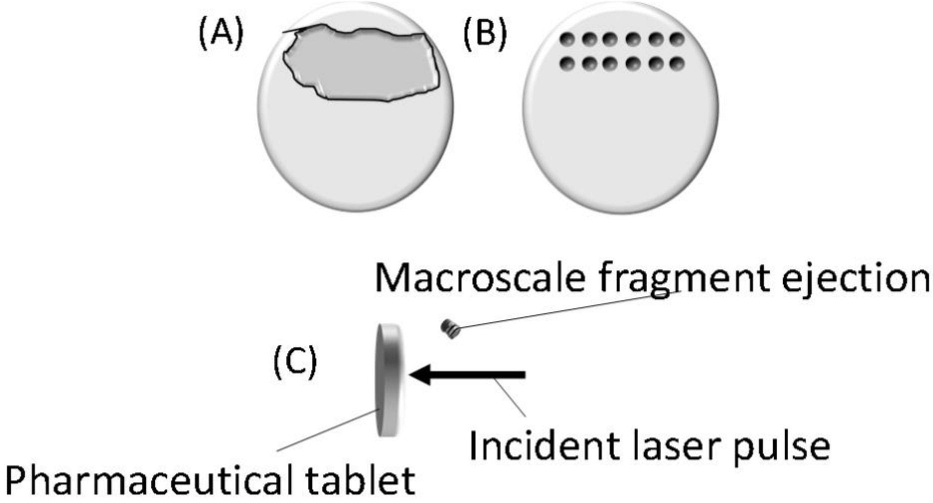


Figure 2.13: Fragment ejection, (A) large delamination of the sample after probing with too much energy. (B) Craters when the correct amount of energy is used. (C) Diagram of macroscale fragment ejection.

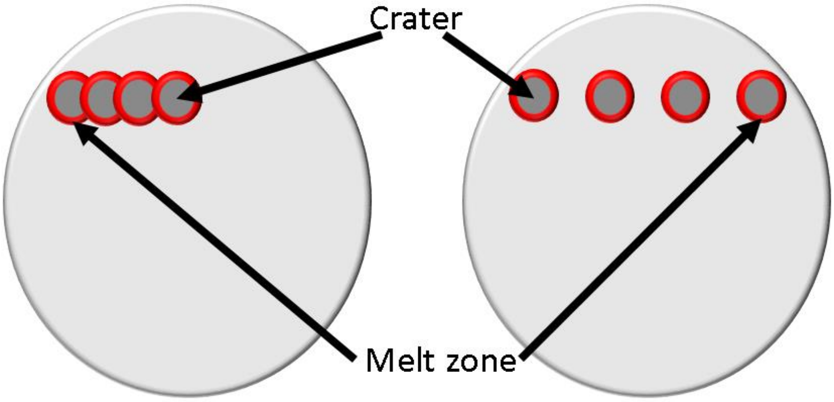


Figure 2.14: Illustration of melt zones. (A) Shots spaced too close together. (B) Shots spaced apart far enough so there is no overlap of melt zones.

A combination of selecting the correct laser energy such that large portions of the sample are not blown out, and spacing shots far enough apart so that melt zones are not probed, allows for optimal probing of these somewhat delicate targets.

To find the best position for the sample, relative to the optical axis of the spectrometer, the samples were positioned at the previously mentioned zero position. Then a variety of target positions around this, relative to the optical axis, were investigated. The goal was to find two positions with similar spectra, so that the non-flat sample surfaces would not overly impact the intensity of the recorded spectra.

Figures 2.15 – 2.17 show waterfall plots of spectra I took at various distances from the optical axis. Figure 2.15 is a rough scan from 0.3 mm to 2.3 mm, where a decline in line intensity is observed as the distance from the optical axis increases. Two finer ranges were investigated, as they appear to hold similar spectra within their respective ranges. Figure 2.16 is a fine scan between 1.3 and 1.8 mm, where the intensity does not vary greatly but was unsuitably low and noisy. Figure 2.17 is a fine scan between 0.3 and 0.8 mm, where the intensity is again relatively stable, but with higher counts than in figure 2.16.

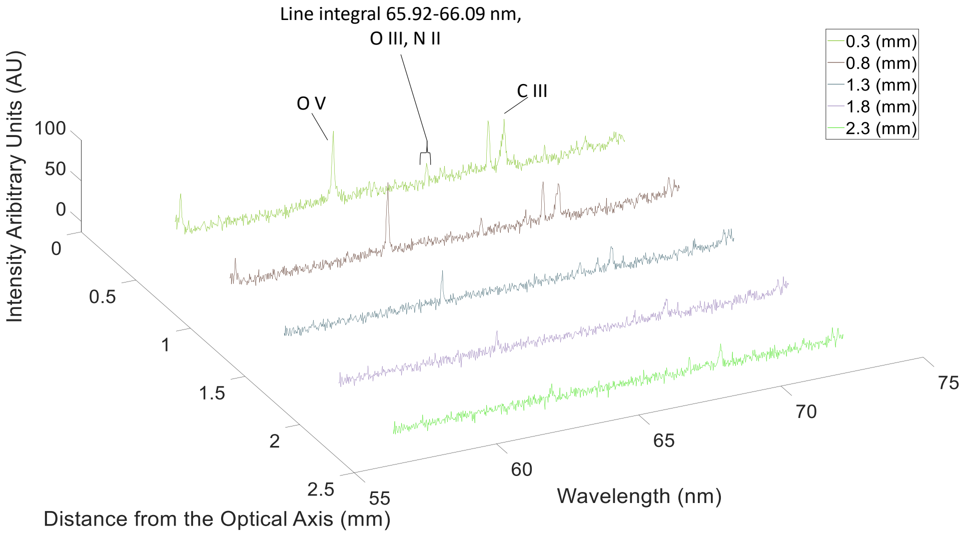


Figure 2.15: Spectra recorded for a rough scan, where the distance from the target surface to the optical axis of the spectrometer is between 0.3 mm and 2.3 mm.

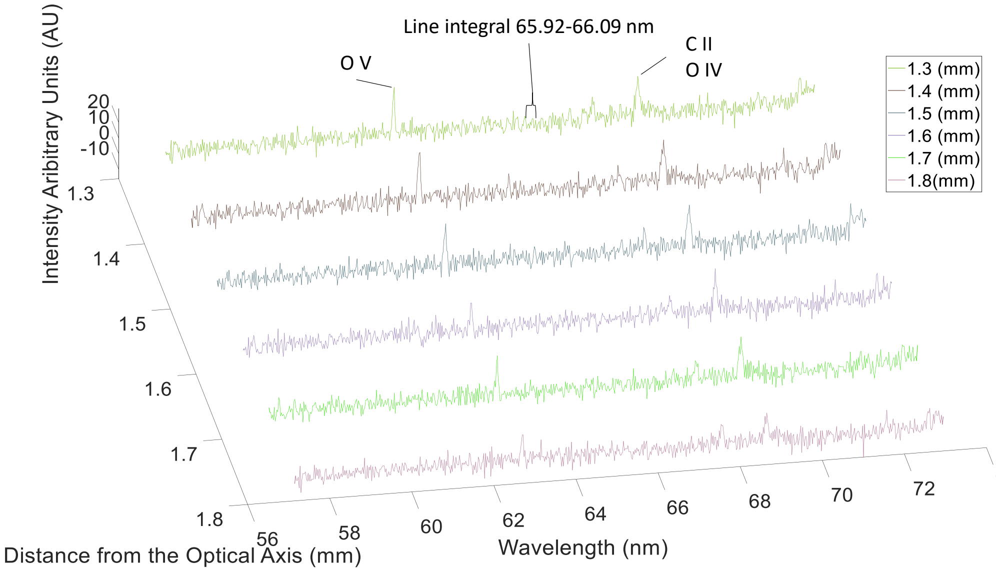


Figure 2.16: Spectra recorded for a finer scan, where the distance from the target surface to the optical axis of the spectrometer is between 1.3 mm and 1.8 mm.

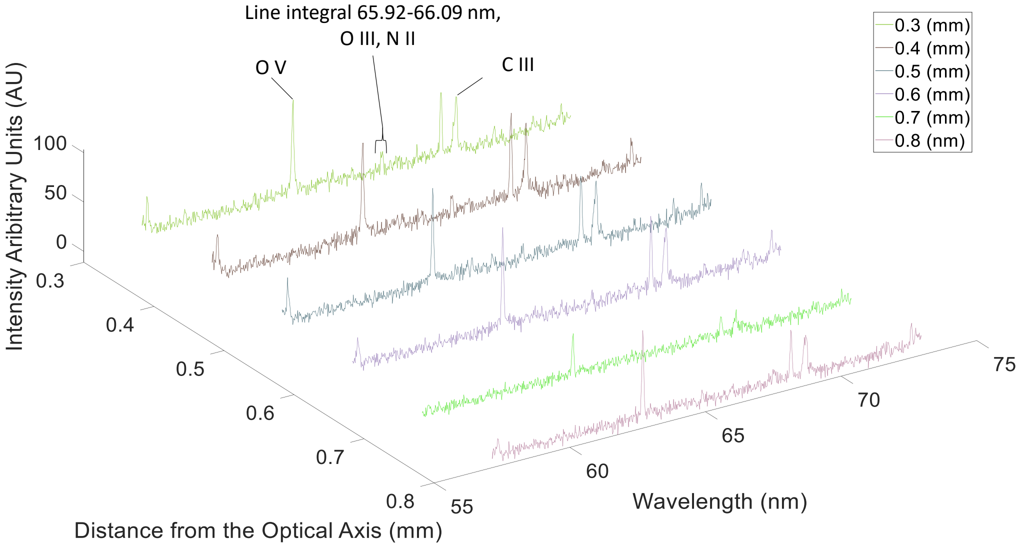


Figure 2.17: Spectra recorded for a finer scan, where the distance from the target surface to the optical axis of the spectrometer is between 0.3 mm and 0.8 mm.

By eye a number of potential sample positions are possible. To aid the decision a number of line integrals were taken and plotted. An example of one of these is shown in figure 2.18. From the graphs the positions 0.3 and 0.4 mm from the optical axis were determined to be the most similar, 0.4 mm from the optical axis was selected for investigation.

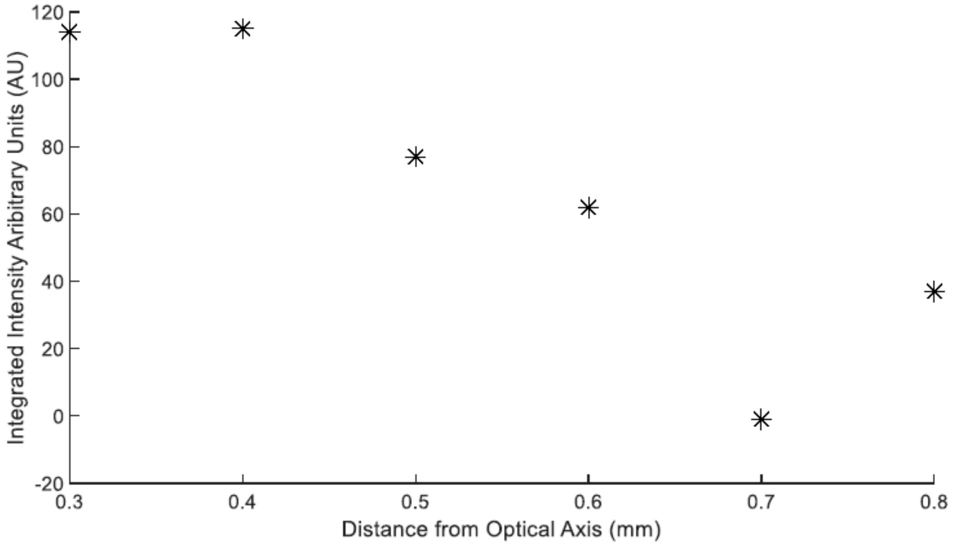


Figure 2.18: Example of the integral of a spectral line in spectra recorded with a distance of 0.3 to 0.8 mm between the sample surface and the optical axis of the spectrometer.

With this the determination of the experimental conditions for the results presented in chapter 5 was complete: The laser energy was reduced to 260 mJ, a 1 mm separation was introduced between plasma formation sites and the sample was moved closer to 0.4 mm from the optical axis.

To ensure reproducible conditions between different experiments the following procedure was used. For each experiment a pharmaceutical tablet was affixed to the target holder in the target chamber. The XYZ manipulator was adjusted to account for the difference is sample thickness (Z) and was varied until vertical and horizontal (XY) limits were noted to probe the flat face of the tablets. Using these limits, 1 mm spaces between shots were planned out. The laser energy was set to the correct value and the spectra were recorded with the camera at -40°C using background subtraction and full vertical binning in the software.

2.3 Pharmaceutical samples

The five tablets used in this study are shown in figure 2.19. These are over the counter products and no sample preparation took place for the study. The five samples are:

1. Disprin Extra Strength (500 mg), a brand of aspirin. The ingredients contained were aspirin (acetylsalicylic acid) as the active ingredient, calcium carbonate, maize starch, citric acid anhydrous, talc (sterilised), sodium laurilsulphate, saccharin, crospovidone and lime flavour.
2. The sweet used in this study is a Silvermints brand mint. The ingredients were sugar, glucose syrup, gelatine, natural mint flavouring and xanthan gum.
3. Panadol 500 mg Soluble Tablets, a brand of paracetamol. The ingredients were paracetamol (500 mg), saccharin sodium, citric acid (anhydrous), sorbitol powder (E420), sodium hydrogen carbonate, sodium laurilsulfate, anhydrous sodium carbonate, povidone and dimeticone.
4. Panadrex Pure 500 mg, a brand of paracetamol. Paracetamol (500 mg) is the active ingredient, with microcrystalline cellulose, maize starch, magnesium stearate.
5. The multivitamin used in this study was the Tesco Health A-Z multivitamins and minerals. The ingredients were Calcium Carbonate, Magnesium Oxide, Vitamin C, Maltodextrin, Dicalcium Phosphate, Cellulose, Ferrous Fumarate, Hydroxypropylmethylcellulose, Modified Maize Starch, Vitamin E, Niacin, Zinc Oxide, Cross-linked Sodium Carboxymethylcellulose, Magnesium Stearate, Silicon Dioxide, Pantothenic Acid, Colours (Titanium Dioxide, Iron Oxide), Manganese Sulphate, Acacia, Talc, Sodium Tetraborate, Maize Starch, Mannitol, Sugar, Cupric Sulphate, Hydroxypropyl Cellulose, Vitamin B6, Thiamin,

Riboflavin, Vitamin A, Triglycerides, Antioxidants (DL-Alpha-Tocopherol, Tartaric Acid), Folic Acid, Chromium Chloride, Potassium Iodide, Sodium Molybdate, Sodium Selenite, Vitamin K, Biotin, Vitamin D, Vitamin B12.

Disprin
(Asprin)

Sweet
(Silvermints)

Panadol
(Paracetamol)

Panadrex
(Paracetamol)

Multivitamin
(Tesco Health A-Z multivitamins and minerals)

Figure 2.19: All tablets used in this thesis, starting from the top left: Disprin, Silvermints, Panadol, Panadrex and Tesco Health A-Z multivitamins and minerals.

2.4 Summary

In this chapter the experimental setup, and the required equipment, needed to produce the spectra for analysis in chapters 4 and 5 have been described. The laser, spectrometer and CCD have been introduced and described individually and the linking of all into a single experimental set-up has been explained. The key considerations for optimization for VUV-LIBS for the pharmaceutical samples used in this study were outlined. These included the requirement for:

i. Sampling on the flat face.

ii. Sampling from the edge further away from the entrance slit and moving inwards horizontally so the melt from previous shots does not obscure the plasma.

iii. Using a lower energy to not blow macro scale ejecta from the samples, allowing for more of the target surface to be probed in a consistent manner.

iv. Keeping a 1 mm separation between plasma formation sites the surface of the sample so the melt zones of previous plasmas are not sampled.

References

[1] Khater MA. Spectroscopic investigations of laser-produced steel plasmas in the vacuum ultraviolet (Doctoral dissertation, Dublin City University); 2001.

[2] O'Leary EP. VUV laser-induced plasma spectroscopy for low level sulphur detection in steel (Doctoral dissertation, Dublin City University); 2007.

[3] Jiang X. Dual-pulse laser induced breakdown spectroscopy in the vacuum ultraviolet with ambient gas: spectroscopic analysis and optimization of limit of detection of carbon and sulfur in steel (Doctoral dissertation, Dublin City University); 2012.

[4] Lu H. UV and VUV photoionization of some metal atoms and ions (Doctoral dissertation, Dublin City University); 2021.

[5] Neettiyath A, Alli MB, Hayden P, Vasa NJ, Sarathi R, Costello JT. Vacuum ultraviolet laser induced breakdown spectroscopy for detecting sulphur in thermally aged transformer insulation material. Spectrochimica Acta Part B: Atomic Spectroscopy 2020; 163:105730.

[6] Continuum, Operation and Maintenance Manual for Surelite Lasers, 2015.

[7] Arieli R. Q (Quality) Factor [Internet]. 1997 [cited 12/08/2019]. Available from: https://perg.phys.ksu.edu/vqm/laserweb/Ch-7/F7s4t2p2.htm

[8] RP Photonics Consulting GmbH. Q Switching [Internet]. 2019 [12/08/2019]. Available from: https://www.rp-photonics.com/q_switching.html

[9] McClung FJ, Hellwarth RW. Giant optical pulsations from ruby. Journal of Applied Physics. 1962 Mar;33(3):828-9.

[10] Smith G. The early laser years at Hughes Aircraft Company. IEEE journal of quantum electronics. 1984 Jun;20(6):577-84.

[11] Degnan JJ. Optimization of passively Q-switched lasers. IEEE journal of quantum electronics. 1995 Nov;31(11):1890-901.

[12] Bollig C, Clarkson WA, Hanna DC. Stable high-repetition-rate single-frequency Q-switched operation by feedback suppression of relaxation oscillation. Optics letters. 1995 Jun 15;20(12):1383-5.

[13] Spühler GJ, Paschotta R, Fluck R, Braun B, Moser M, Zhang G, Gini E, Keller U. Experimentally confirmed design guidelines for passively Q-switched microchip lasers using semiconductor saturable absorbers. JOSA B. 1999 Mar 1;16(3):376-88.

[14] Paschotta R. Intensive light pulses, tailored to your needs [Internet]. 2005 [cited 13/08/2019]. Available from: https://www.rp-photonics.com/qs_laser_d.pdf

[15] Hellwarth RW. Control of fluorescent pulsations. In Advances in Quantum Electronics 1961 (p. 334).

[16] RP Photonics Consulting GmbH. Q Factor [Internet]. 2019 [12/08/2019]. Available from: https://www.rp-photonics.com/q_factor.html

[17] Burle Industries lnc. GCA Product data sheet [internet]. 2006 [cited 13/08/2019]. Available from: http://www.burle.com/cgi-binlbyteserver.pllpdf/tp184.pdf

[18] Collimated Hole Inc Capillaries and Capillary Structures [internet] 2019 [cited 13/08/2019]. Available from: http://www.collimatedholes.com/products.html

[19] Acton Research Corporation, Model VM-521, Operation manual, 1990.

[20] Rowland HA. LXI. Preliminary notice of the results accomplished in the manufacture and theory of gratings for optical purposes. The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science. 1882 Jan 1;13(84):469-74.

[21] Rowland HA. XXIX. On concave gratings for optical purposes. The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science. 1883 Sep 1;16(99):197-210.

[22] Beutler HG. The theory of the concave grating. JOSA. 1945 May 1;35(5):311-50.

[23] Namioka T. Theory of the concave grating. I. Josa. 1959 May 1;49(5):446-60.

[24] Samson JA. Techniques of vacuum ultraviolet spectroscopy. John Wiley & Sons, 1967.

[25] Andor Technology: Scientific Digital Camera Solutions, 2006.

[26] Quantum Composers, Inc: 9500 Plus Series & 8510 Board Level Pulse Generator Operating Manual V1.3, 2011

3 Machine learning for LIBS

This chapter outlines the theory behind machine learning and the implementations I used for classification in this thesis. The corresponding code for each application can be found in appendix C. A brief introduction is given in section 3.1. The various machine learning techniques used in this work, and their underlying theory, is detailed in section 3.2. Section 3.3 will outline the processing of the spectra required to apply the convolutional neural networks (CNNs), while the algorithms/framework implemented to present the machine learning techniques with spectra and extract results are described in section 3.4.

3.1 Introduction

Multivariate analysis, in the form of principal component analysis (PCA), was detailed in section 1.5.1. The aim of this multivariate analysis was to find a representation of the many variables, both dependent and independent, which may or may not be known and to determine coefficients to explain observed variance in the experimental spectra. This differs from machine learning as applied in this thesis, which is used for only classification analysis and not the analysis of many variables. This is a whole spectrum approach where, instead of looking for the underlining variables and their coefficients, machine learning is used to recognise complex patterns in the experimental spectra. In effect the causes of the spectral variations lose importance and the net spectral changes take primacy. The goal is then to teach the machine learning algorithm to associate different spectral features with different sample types and to test to see how it well it correctly classifies a set of spectra that is unseen during training.

There are two main types of machine learning, supervised and unsupervised learning. Supervised learning, such as CNNs and support vector machines (SVMs), is when the network has knowledge of a samples class during training. Unsupervised learning, such as competitive learning (COMP) and self-organizing maps (SOMs), is when the network has no knowledge of a samples class during training. An advantage of supervised learning is that any issues can be identified at an early stage of the training, and user input can assist the training, but prior knowledge of what is being tested is needed. Unsupervised training is generally easier to implement and does not require any specialised user knowledge, but any issues cannot be identified until training is complete and testing begins.

Machine learning is applied in this work with the sole aim of distinguishing between different pharmaceutical tablets. The application of four machine learning techniques has been investigated. COMP, which is a linear arrangement of neurons [1-18]; SOMs, which is a topology of neurons [19-29]; SVMs, which is classification based on dimensional reduction and projection [30-40]; and CNNs, which are complex layers of neurons consisting of image processing and signal processing and ultimately distinguish between different classes of input data [41-49].

Following an initial literature review, the machine learning techniques used in this work were selected for their demonstrated ability to classify data. The addition of each technique brought new knowledge to enrich the study as a whole. COMP is a primitive technique with long computation times, but simple in application. SOMs bring a discussion of selecting the right number of neurons to achieve the best classification possible for a given application, building in complexity but leading to a reduction of computational time. SVMs were a challenging addition as they require inputs to be 2 data points long, so PCA was used to dimensionally reduce each spectrum to two PCs. SVMs also introduced the concept of Bayesian optimisation, a technique that would later provide an automated way to tune the hyperparameters of a CNN far more efficiently than manual parameter tuning. As will be seen in chapters 4 and 5 the CNNs, by using a 2D reconstruction of the entire spectrum as detailed in section 3.3, consistently bring the best average classification accuracy to the table.

In the context of the work in this thesis, the decision of how similar certain analytes are is outsourced to a machine, leveraging its fast compute time. In general terms machine learning is pattern recognition by iterative transcription. Weights are transcribed closer to inputs over iterations, those weights then learn those input patterns for classification [50].

A motivation for offloading the analysis to a machine is an economic one, in terms of time and expertise required: if datasets are sufficiently large it can become unfeasible for an individual (who may require many years specialist spectroscopy training) to analyze many spectra in a timely manner. This issue, in general, has led to the emergence of fields such as “big data”, where machine learning has grown in prominence due to its capabilities in pattern recognition. Another useful benefit of machine learning is its ability to abstract, which proves valuable when subtle differences between spectra from different samples are not obvious (or when complex differences between spectra cannot be found using uni- or multivariate techniques). For example, where one cannot find an isolated line, lines or other spectral features that are clearly unique to a particular sample machine learning can, through learning, start to distinguish and emphasize minor differences. As the learning progresses the network becomes more sensitive to these minor differences, such that the network is trained to classified best on this feature. This ability to abstract and learn differences will be demonstrated in section 5.5.3, where two brands of paracetamol with largely the same spectral signature and lines present were distinguished from one another to a very high degree using machine learning.

3.2 Machine learning techniques

As machine learning has become more popular the variety of techniques has dramatically increased. These differ in a number of ways, such as how the data is presented to the machine, how the weights are updated and optimized and how the results are presented. Four types were chosen for study in this thesis, and the details of these will be given in the following sections.

3.2.1 Competitive learning

COMP, as investigated in this work, is a linear arrangement of neurons that consist of an input layer and competitive layer, and the outputs of the competitive layer denotes which neuron a spectrum has clustered to [51]. A visualization of COMP is shown in figure 3.1.

COMP is learning where neurons compete to with one another to cluster input spectra onto themselves. The mechanism by which learning takes places is called *winner-take-all* [52-60] – the neuron that wins the competition has the input spectrum cluster to it and its weights are updated. The neurons effectively act as bins. As more data is fed to the network the neurons become specialized feature detectors.

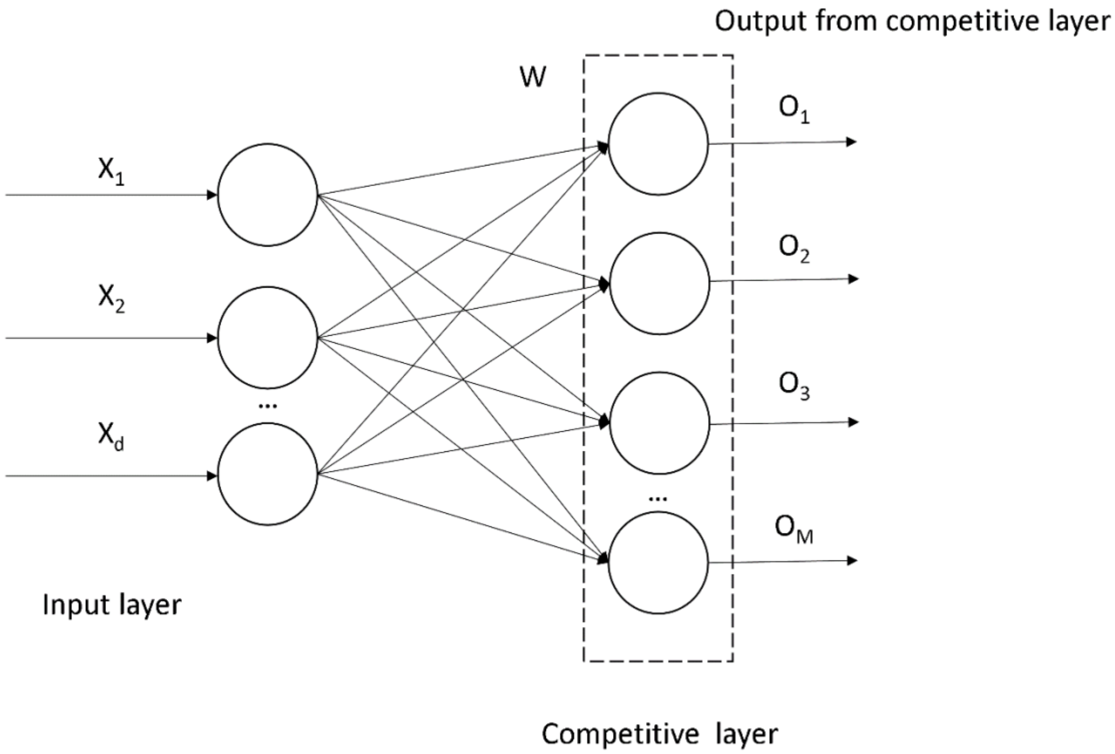


Figure 3.1: Competitive neural network architecture [51].

The closeness or similarity of weights, w_i, and inputs, x_i, is measured using the Euclidean distance, as shown in equation 3.1 [61]. The smaller the distance the more similar a neurons weights are to an input. The weights are initialized to the average value of the inputs and the weights are updated after each iteration as laid out in equation 3.2 [62], where w'_i is the new weight, w_i is the old weight and α is the learning coefficient.

$$Euclidean\ distance = \sqrt{\sum_{i=1}^{N}(x_i - w_i)^2} \qquad (3.1),$$

$$w'_i = w_i + \alpha\,(x_i - w_i) \qquad (3.2).$$

The learning coefficient initially has a value of one and gradually reduces towards zero, this means there are larger weight changes at the beginning and more gradual changes at the end of training. This process is repeated until the weights become more like the input patterns that cluster on them, thus becoming feature detectors. The learning can end either after a set number of iterations or once a threshold of similarity between the inputs and the weights has been reached.

For the current application of COMP the inputs in figure 3.1, x_i, are normalized spectra (with the length given by the number of pixels in a spectrum). Each of the neurons have an associated weight, where the number of neurons is equal to the number of different classes present, e.g. two neurons for 1 vs 1 testing where there are two expected classes. Each of the weights, w_i are initialized to the average of the training inputs, therefore each weight has the same length as the inputs, which with normalized spectra limits their initial values to between -1 and +1. As each test spectrum is presented to the COMP network the neuron that best matches the test spectrum has its weights updated. The best match is determined by equation 3.1, and the weights updated by equation 3.2. Once the unsupervised training is complete each neuron should be trained to recognize one class, as its weight should best match a test spectrum from an individual class. The number of outputs, O_m, from this network will be equal to the number of neurons, i.e. the number of classes under consideration. These will all be zero except for the neuron corresponding to the predicted class from the model [52]. If the predicted class corresponds to the actual known class then the COMP network is successful, if not then misclassification has occurred.

3.2.2 Self-organizing maps

SOM is a more advanced version of COMP. While COMP updates a single "winner-take-all" neuron, in SOMs all neurons within an update radius are updated at the end of an iteration.

Another distinction between SOMs and COMP is that SOMs have neurons that are traditionally arranged like a surface. This spatial arrangement acts like an expansion of feature space while allowing an update radius to act as a mechanism for clustering. Figure 3.2 shows a 5x5 SOM, centered at the 13th neuron. N13(1) has an update radius of 1 and N13(2) has an update radius of 2.

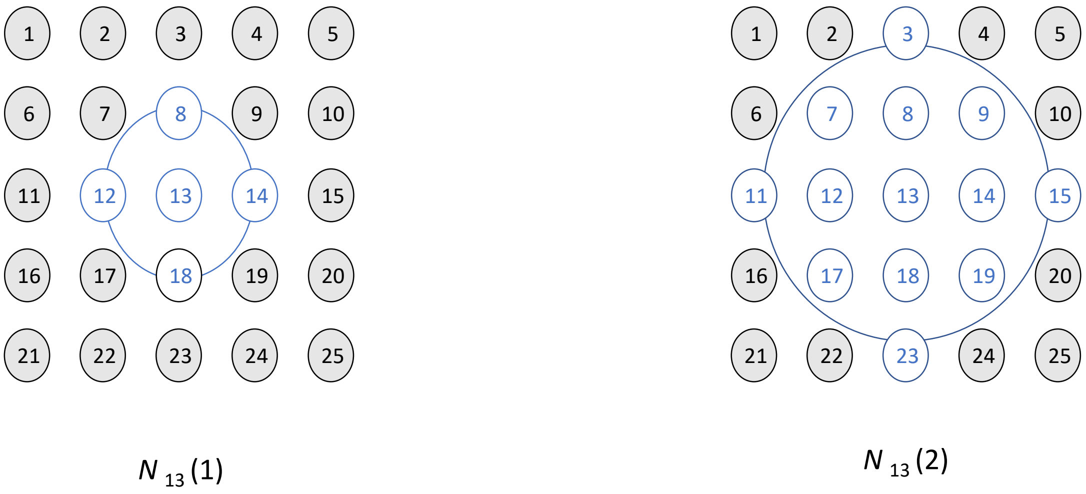


Figure 3.2: Neural network architecture of a 5x5 neuron self-organizing maps [63].

The winning neuron is identified using the same method as in COMP, i.e. the neuron with the shortest Euclidean distance to the input. In SOMs all the inputs are presented to the network before any weights are updated. The winning neuron for each input is calculated. In SOMs the weights, w, are initialized as

$$w = \mathrm{repmat}(\mathrm{inputMean}, 1, \mathrm{numPos}) + (\mathrm{PosBasis} \times \mathrm{pos}) \qquad (3.3),$$

where the first term, $\mathrm{repmat}(\mathrm{inputMean}, 1, \mathrm{numPos})$, consists of a vector of repeating numbers, of length numPos, and the repeating number is the mean of the input spectra, $\mathrm{inputMean}$. These variables are used as inputs for the MATLAB function repmat [64] which generates the matrix containing the repeating value.

The second term is the multiplication of pos and $\mathrm{PosBasis}$. pos is a min-max mapping function to map the initial position given by equation 3.4, determined when the network is created, between -1 and 1. The positions with the minimum value will be converted into -1 and the ones with the maximum value will be converted to +1. The rest will fall between -1 and +1. In equation 3.4 $minPos$ is the minimum position of $tripos$ and $difPos$ is the difference between the maximum and minimum position values of $tripos$. ($tripos$ is a range of values initialized using the hextop function [65] in MATLAB, which generates a grid of equilateral triangles with sides of length 1.)

$$pos = 2 \times ((tripos - minPos)./difPos) - 1; \qquad (3.4).$$

Figure 3.3 (where half step shifting is used in the x-axis horizontal placement from one row to the next) and table 3.1 (4 decimal places) show an example of the points generated for a 3x4 neuron SOM.

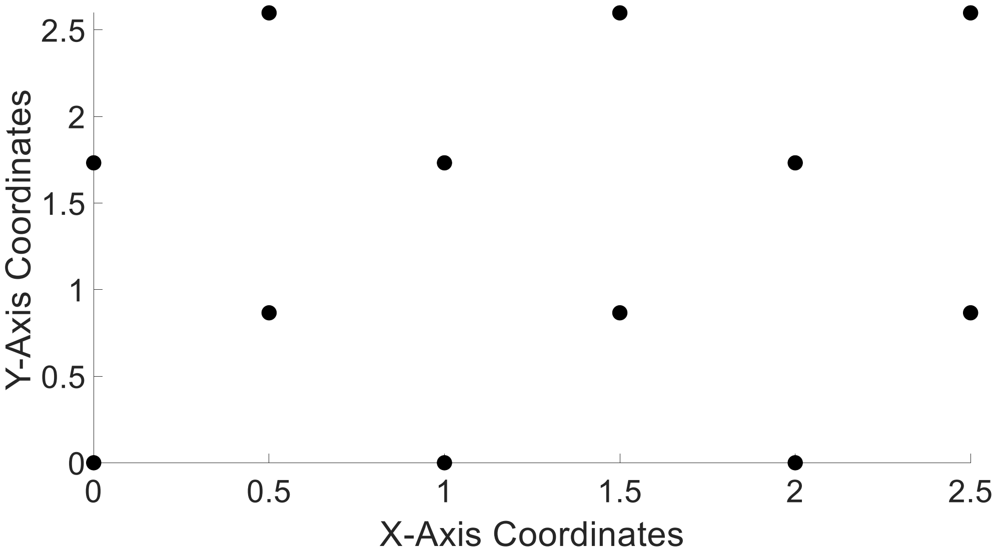


Figure 3.3: Visualization of positions initialized by the hextop function.

The Pythagorean theorem is used to calculate the appropriate y-axis vertical offset, e.g., $\sqrt{0.5^2 + 0.8660^2} = 1$, which is where the value 0.8660 (rounded to 4 decimal places) comes from in table 3.1 and its multiples are used for further vertical offset, as seen in figure 3.3.

Singular value decomposition (SVD) is a methodology to extract eigenvectors and is often utilized for dimensional reduction techniques such as principal component analysis (PCA). $PosBasis$ is the SVD basis defined by equation 3.5

$$PosBasis = U \times S \times 2.5 \times stdev(V) \quad (3.5),$$

where U contains the eigenvectors of the matrix AA^T (A is an input array), V contains the eigenvectors of the matrix A^TA, and S contains the singular values which are non-negative square roots of the eigenvalues of the matrix A^TA [66]. The factor of 2.5 multiplied by the standard deviation of V spreads the weight positions.

Table 3.1: Position coordinates generated by the hextop function.

X-axis coordinates	Y-axis coordinates
0.00	0
1.00	0
2.00	0
0.50	**0.8660**
1.50	0.8660
2.50	0.8660
0.00	1.7321
1.00	1.7321
2.00	1.7321
0.50	2.5981
1.50	2.5981
2.50	2.5981

In this case experimental spectra are used as inputs, held in the array A such that each row in the array is a spectrum. Before U, S and V are calculated by SVD the input spectra held in matrix A are mean centered, this entails subtracting the mean value of a spectrum from every element in said spectrum in that row.

The weight vectors are updated such that they approach the average values of inputs. The inputs that are averaged are those that correspond to the winning neuron and those that fall within the neighbourhood of a winning neuron, as dictated by the update radius.

The initial update radius is set to 3 and decays by 1 until the radius reaches a size of 1 [63]. This process is in effect quite analogous to clustering techniques that are more familiar in physics and engineering such as k-means clustering [67-69] or the k-nearest neighbours' algorithm [70], where data belonging to the same class is clustered together in an iterative fashion. The radius must be of appropriate size to achieve separation between classes – too small and it will not update an area large enough, too large and interclass separation is lost due to unwanted grouping between the classes.

To look at this theory in practice we can see how 48 spectra have clustered and been grouped onto a SOM, shown in figure 3.4, after completion of the training. The SOM in this case is a 4x4 neuron arrangement containing 16 neurons in total. In the training set there were 12 spectra per class and 4 classes, with each class corresponding to a distinct sample type. The training procedure has grouped each of the training spectra to a neuron with the most similar weight, where each neurons weight is the same length as an input spectrum. The number of spectra after training grouped to each neuron is shown by the size of the purple hexagons (and the number within) in figure 3.4. Ideally each class would be clustered close together or even to just a single neuron. To then test the network a spectrum would be presented, and the output of all neurons would be zero, except for the neuron which is closest to the test spectrum. If the predicted class corresponds to the actual known class, then the SOM network is successful, if not then misclassification has occurred.

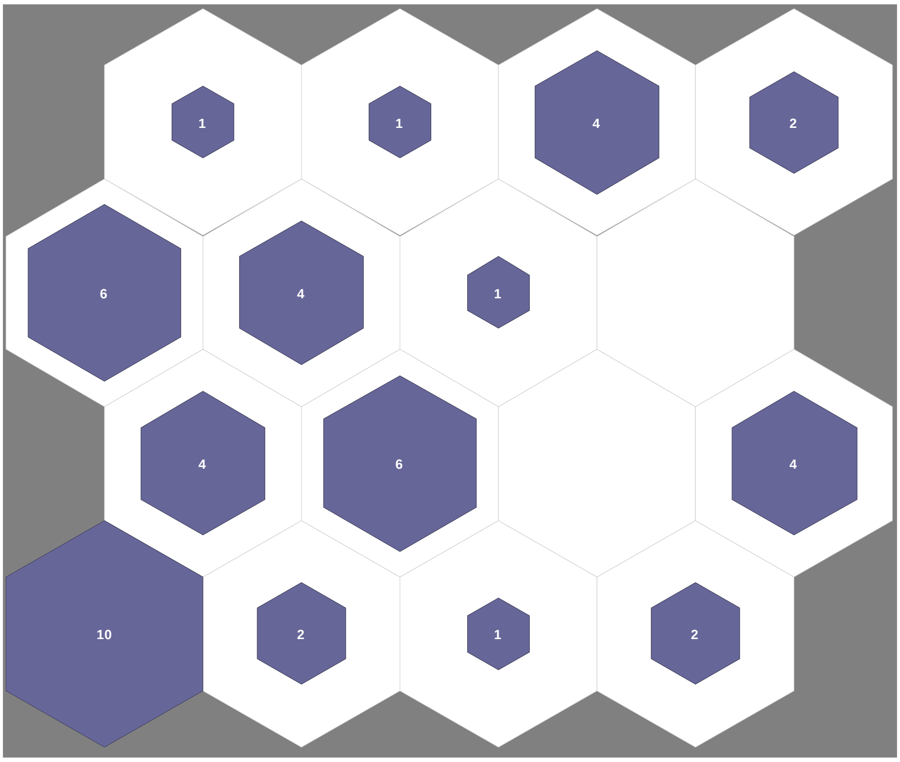


Figure 3.4: A 4x4 SOM with 48 individual spectra clustering onto neurons.

Table 3.2: Results of example neuron clustering.

	Neuron number clustered
Sample 1	3,5,6,11
Sample 2	9,10,13,14
Sample 3	4,8,15,16
Sample 4	1,2

At first the results and visualisation shown in figure 3.1 may appear disappointing and far from ideal. The observed clustering does, however, present an important finding that can be leveraged. Each of the classes did cluster to many neurons instead of just one, but the set of neurons registered by each of the classes is unique, as shown in table 3.2. In effect the network is telling us that each class consists of subclasses, and the neurons are trained to identify these subclasses. As long as the neurons a particular class are clustered to are unique, then classification can be deemed still possible. When a test spectrum is presented to the network only one neuron will generate a non-zero output. If this corresponds to one of the set of neurons corresponding to the known class, then the SOM network is successful [63]. This knowledge was learned and implemented for classification with SOMs in this thesis. The explicit conditional programming which implements this understanding is shown in the code given in appendix C.

3.2.3 Support vector machines

SVMs are a type of machine learning that rely on separating classes by drawing a hyperplane between them. The points closest to the hyperplane are referred to as support vectors, as illustrated in figure 3.5. The optimal positioning of the hyperplane is determined by achieving the largest possible margins. Margins are the distance between the support vectors and the hyperplane, as illustrated in figure 3.6. Sometimes data is difficult or impossible to separate in its 2D representation, but by projecting in 3D it can become possible to separate classes by a hyperplane, as illustrated in figure 3.7.

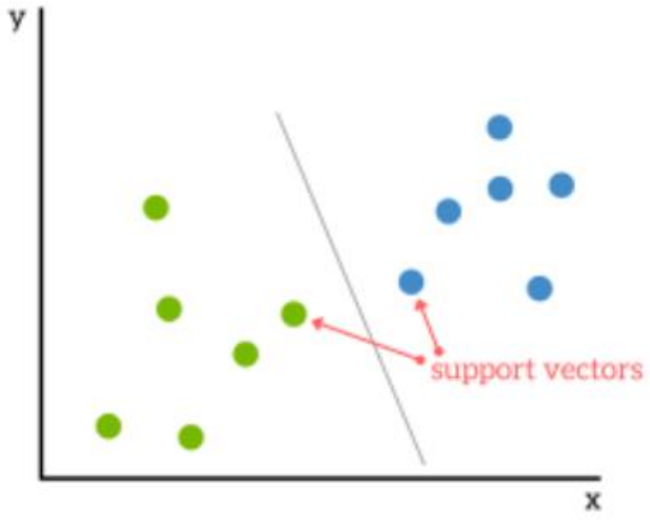


Figure 3.5: Illustration of support vectors and their proximity to a hyperplane separating two data sets [71].

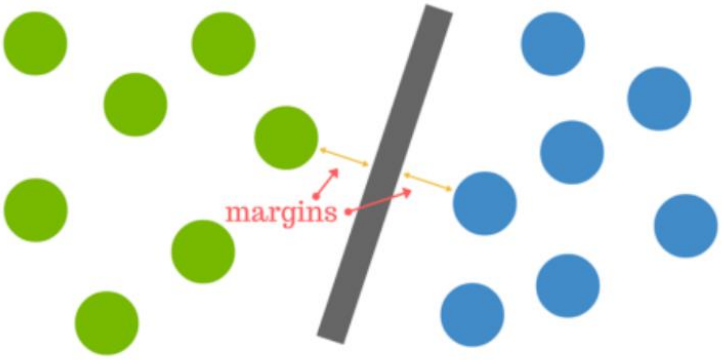


Figure 3.6: Illustration of margins between support vectors [71].

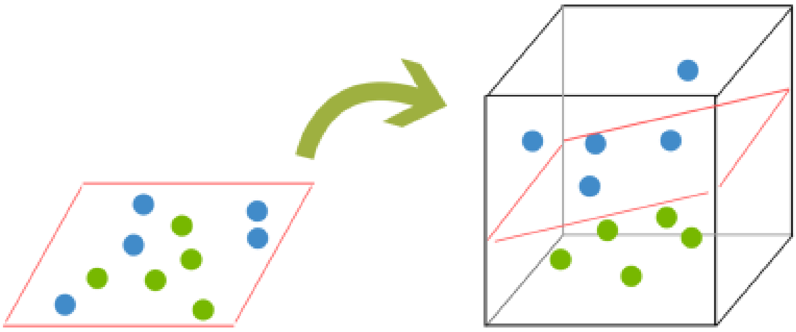


Figure 3.7: Illustration of support vector machines projecting from 2D to 3D [71].

SVM and CNNs are known as supervised learning methods as during training knowledge of what class the training sets belong to is known. This is because SVMs and CNNs both rely on a system of labelling. Data used in training, validation and testing is labelled before it enters the network. The label represents what class the data belongs to, and by comparing a predicted label to a true label it is possible to gauge if a prediction is correct or not. The loss function, otherwise known as the cost function, is a measure of performance of a prediction. It has a value greater than zero when the system does not predict a true positive [71,72].

SVMs and CNNs are considered black boxes. They require a large number of parameters that need to be defined prior to training, with these parameters known as hyperparameters. Human optimization is not fast enough to step through these unknown parameters and change their values at a sufficient rate. Some of these parameters are dependent on one another and some are independent but still have a clear impact on performance of a system.

The solution is machine driven optimization; where the various parameters are changed randomly but attempts to approach a solution with the lowest amount of loss. As loss is how badly a system performs, minimising it means the system has been optimized for performance. This will occur when the largest number of true positives are being predicted [73-78].

Bayesian optimization is used to optimize the hyperparameters for training for the SVMs and CNNs. If f(x) is the loss function, as a function of the hyperparameters x, it is entirely unknown, but can be assumed to be described by a gaussian process (the generalization of a gaussian distribution over functions instead of random variables). The goal of Bayesian optimization is to minimize f(x). This is done by applying and maximizing an acquisition function, called the expected improvement (EI). Equations 3.6-3.8 represent abstract formulae used to represent statistical processes calculated within the black boxes for optimization.

$$EI = [max\{0, f(x) - f(\hat{x})\}] \qquad (3.6),$$

$$EI = \begin{cases} (\mu(x) - f(\hat{x}))\Phi(Z) + \sigma(x)\,\phi\,(Z) & If\ \sigma(x) > 0 \\ 0 & If\ \sigma(x) = 0 \end{cases} \qquad (3.7),$$

$$Z = \frac{\mu(x) - f(\hat{x})}{\sigma(x)} \qquad (3.8),$$

where Φ(Z) and ϕ (Z) are the cumulative distribution and probability density function of the (multivariate) standard normal distribution, respectively, $\mu(x)$ is the expected value of the loss, $f(\hat{x})$ is the current best value and $\sigma(x)$ is the uncertainty around the point x.

The EI is high when $\mu(x)$ is higher than $f(\hat{x})$ and when $\sigma(x)$ is large. This can be explained in simple terms using figure 3.8. When $\mu(x)$ is larger than $f(\hat{x})$, sampling is taking place in a region that is currently optimal, thereby exploiting it. When $\sigma(x)$ is high a region that has not been sampled yet is being explored. This pairing is known as exploration versus exploitation [79].

$$EI = \begin{cases} \boxed{(\mu(x) - f(\hat{x}))\Phi(Z)} + \boxed{\sigma(x)\,\phi\,(Z)} & If\ \sigma(x) > 0 \\ 0 & If\ \sigma(x) = 0 \end{cases}$$

Figure 3.8: EI formula, the red section showcases exploitation, and the green section showcases exploration [79].

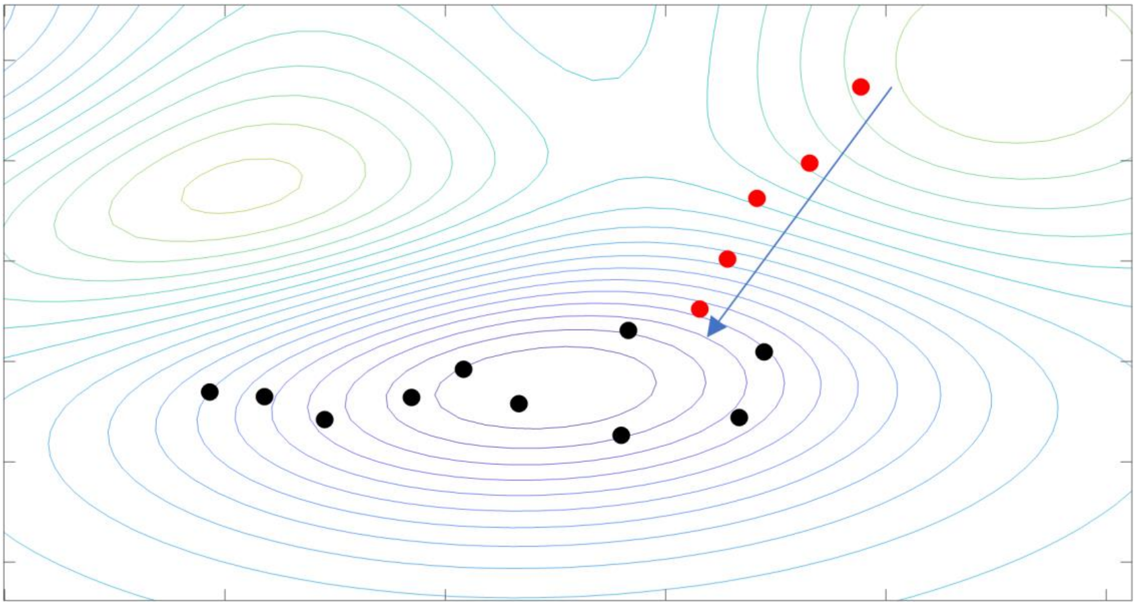


Figure 3.9: Analogy of exploration vs exploitation for Bayesian optimization, exploration towards local minima is illustrated by the blue vector.

In figure 3.9 note how, on approach, the acquisition function explores towards a local minimum (red dots) but once it is within a local minimum it exploits the region (black dots). In the graphical representation this appears similar to a mix of a Bayesian approach with gradient descent. However, in this case it is purely a Bayesian approach and the apparent gradient decent is due to sampling by the EI acquisition function, whose goal is to continue to randomly sample in an exploratory fashion to avoid trapping in a local minima (as opposed to finding the true minimum). Given observed values of $f(x)$, the subsequent values of f are updated by the gaussian process. The new value of x is found by maximizing the acquisition function. This brings us back to the beginning where the new value of x is placed in $f(x)$ and the value of f is calculated, via the gaussian process. This procedure is repeated for several preset iterations or until convergence is achieved.

In practice three pieces of information are passed to the network for each training spectrum, namely the first 2 PC scores and the known class. The black box of the network is then allowed to determine the best values of the hyperparameters during the training step. For testing two pieces of information are passed to the network for each test spectrum – the first 2 PC score. The network then predicts which class a test spectrum belongs to and if this corresponds to the actual known class then the network is successful [72].

3.2.4 Convolutional neural networks

CNNs distinguish themselves from the other machine learning techniques in this work because they accept image inputs, not spectra in the form of vectors. Historically the driving force behind the design of CNNs was as an image processing solution and much of the early and continuing work are focused on this aspect. To take advantage of existing techniques and available toolboxes the pipeline of processing within a CNN requires an input image for the 2D convolution, where various 2D feature maps are generated from the input image It is these feature maps that act as the feature detectors that will be used as class identifiers. Appendix B details the use of Matlab's proprietary ResNet50 network. The research leading to this thesis began with transfer learning, i.e., training this premade network architecture originally designed for one application and applying it to another application. The transfer learning approach of using a premade network came with severe drawbacks. ResNet50 had 177 layers and required inputs of 224x224 pixel images, with a memory requirement of over 9.6 GB on a graphics processing unit (GPU). This large memory requirement limited the scope of the studies that could be conducted as scaling to networks that needed larger datasets for training, such as the data augmentation study in section 5.1.1, required even larger resources.

The solution was to design a purpose-built network. The main considerations for this network was good classification results with as few layers as possible and with as small an input image size as possible, as both the number of layers and the input image size significantly contribute to the memory consumption of the network. A trial and error approach resulted in the 7-layer network outlined in figure 3.10, which met the design requirements (good classification with as few layers as possible). This ad-hoc network also required smaller input images and the classification performance match the ResNet50 network, as shown in appendix D.

The image input layer takes in 50x50 pixel 2-D images and applies data normalization [80]. The 2-D convolution layer applies a 5x5 sliding filter with a stride of 1 on the input image. Stride is the amount the filter is translated by before it operates again. A Frobenius inner product is taken between the input and the weights of the filter, an example to illustrate this process is shown in figure 3.11.

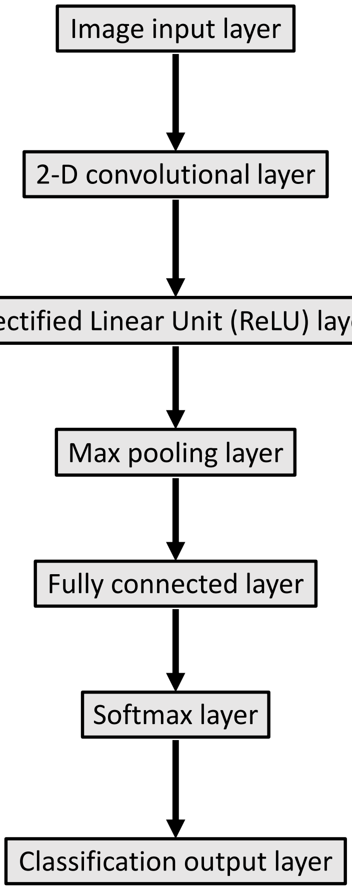


Figure 3.10: Architecture of the 7 layer CNN and the sequential processing passing from layer to layer.

$$A = \begin{bmatrix} A_{11} \dots A_{1n} \\ A_{21} \dots A_{2n} \\ \vdots \\ A_{n1} \dots A_{nn} \end{bmatrix} \qquad B = \begin{bmatrix} B_{11} \dots B_{1n} \\ B_{21} \dots B_{2n} \\ \vdots \\ B_{n1} \dots B_{nn} \end{bmatrix}$$

$$Example\ where\ A = \begin{bmatrix} 1\ 2\ 3 \\ 4\ 5\ 6 \end{bmatrix} \text{ And B} = \begin{bmatrix} 2\ 3\ 4 \\ 6\ 1\ 1 \end{bmatrix}$$

$$\langle A, B \rangle_F = (1\times2)+(2\times3)+(3\times4)+(4\times6)+(5\times1)+(6\times1) = 55$$

Figure 3.11: Example of a Frobenius inner product between two 3x3 matrices.

Each resulting output from the 2-D convolution layer populates a pixel in a 46x46 feature map. In this application 20 filters are applied, resulting in the extraction of 20 feature maps, each being 46x46 pixels. Finally, a bias term is added [81]. Learning occurs when the weights extract features in the resultant feature maps useful for classification. Optimization techniques such as Bayesian optimization, described in section 3.2.3, and stochastic gradient descent, described later in this section, optimize the hyperparameters that adjust the values of weights and biases for each iteration.

The ReLU layer performs a threshold on each element of the input, setting it to zero if its value is less than zero [82]. The threshold is equivalent to the conditions laid out in equation 3.9:

$$g(x) = \begin{cases} x, & x \geq 0 \\ 0\,, & 0 < 0 \end{cases} \qquad (3.9).$$

The max pooling layer down samples the input by dividing it into rectangular regions and computing the maximum value of those regions. Those maximum values are then stitched beside each other as the output of this layer [83]. The max pooling in this study uses 2x2 filters and a stride of 2. This process down samples the input feature maps from 20 individual 46x46 pixel feature maps to 20 individual 23x23 pixel feature maps. A visual example of this process for a single feature map is shown in figure 3.12.

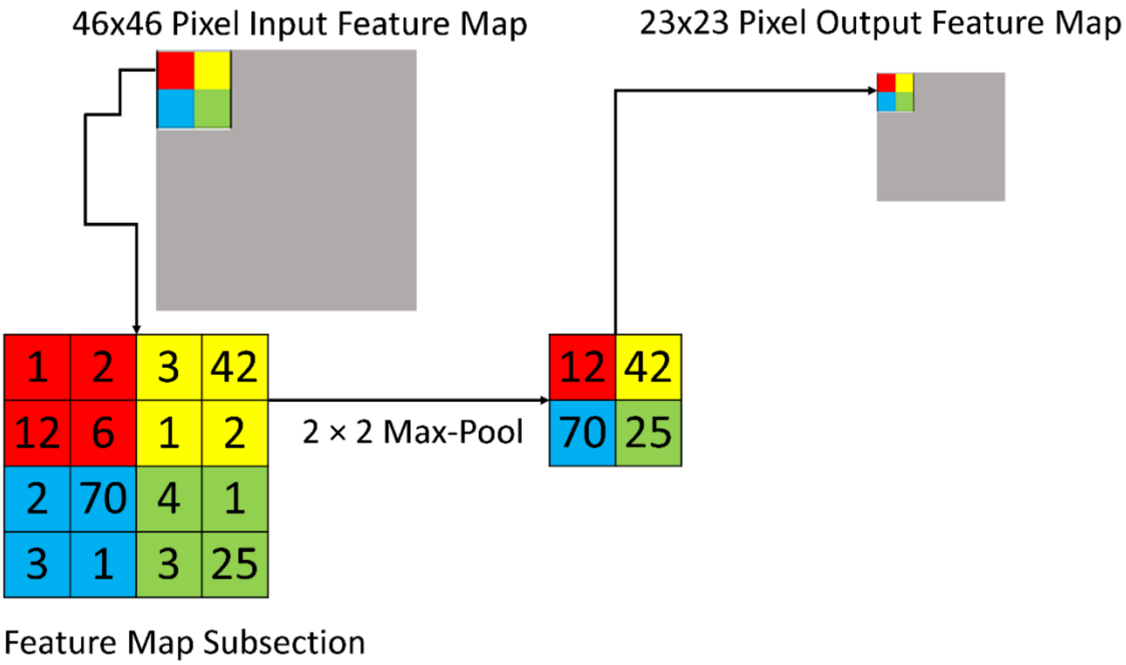


Figure 3.12: Example illustrating the down sampling that occurs in max pooling [84].

In the fully connected layer, the input is flattened and is multiplied by a weight matrix (1x10580) and a bias vector is then added [85]. As the output size of this layer was set to the number of classes, a weight matrix of dimensions (number of classes × 10580) is generated and biases are generated equal to the number of classes. The weights of the fully connected layer and the 2-D convolution layer were initialized using Xavier initialization [81,85]. In Xavier initialization weights are determined by a uniform distribution, with a mean of zero and a variance, σ, as described in equations 3.10 and 3.11. This distribution is symmetric about zero, with a starting point $(-\sqrt{b})$ and an end point $(\sqrt{b})$.

$$\sigma = \frac{1}{12} \times (2b^2) \qquad (3.10),$$

$$b = \sqrt{\frac{6}{number\ of\ inputs + number\ of\ outputs}} \qquad (3.11),$$

For the 2D convolutional layer the number of inputs is 7500, which corresponds to 50 by 50 pixels and 3 RGB channels (50 × 50 × 3= 7500) and the number of outputs is 42320, which corresponds to the 20 46x46 pixel feature maps generated (46 × 46 × 20 = 42320). For the fully connected layer the number of inputs is 10580 and the number of outputs equals the number of classes.

At this point it is worthwhile to recall the purpose of this ad hoc network: to provide good classification results while significantly reducing the memory consumption and runtime required. While the number of outputs from the fully connected layer will always equal the number of classes, the number of inputs will be determined by choices made earlier in the workflow. For example, increasing the size (from 5x5) and stride (from 1) for each of the 2-D convolution layers (or decreasing the number of filters) will decrease the size of the output from each (46x46 as described above), which would lead to a lower memory requirement. In a similar manner changing the size (from 2x2) and stride (from 2) in the max pooling layer can lead to less (currently 20) and smaller (currently 23x23) outputs from this layer, which are the inputs for the main training in the fully connected layer (currently 10580 [= 20x23x23]).

This seems like a positive as the smaller input would require less memory and run faster. While it may sound initially tempting to down sample as much as possible early on it is important to remember that the feature maps being generated in these layers are the feature detectors in the CNN, and we also require the network to provide good classification results. So providing as much information as possible will assist with this aspect of our aim, but as the information increases the computational cost also increases. This leads to a contradiction in our goal and is where the experience and skill of the network creator comes into play. At the design stage early indicators of critical parameters that lead to unbeneficial increases in resources, or trends in groups of parameters, must be identified and brought back into an acceptable range, whilst not having an undue reduction in network classification performance.

During the early stages of designing the network this approach was undertaken, and improved when required throughout the research period, resulting in the details of the described network. For the purposes of the study presented here this network was deemed suitable, as it approached the classification performance of the ResNet50 network but required a fraction of the resources. ResNet50 typically took eight days to run using over 9.6 GB of memory, while the ad hoc networks runtime was of the order of two minutes using 20 MB of memory. It is quite probable the final network would look dissimilar if designed for a different application. This highlights what is perhaps the major drawback of machine learning – the learning takes placed within a creator defined framework, so despite how well the network can learn any flaws in the framework cannot be overcome.

In the fully connect layer, whose input size are determined by the above considerations, stochastic gradient descent and Bayesian optimization are responsible for tuning hyperparameter values such that the weights and biases are optimal for classification. As more optimal weights and biases are approached classification will improve and that is how learning occurs in this layer. The SoftMax layer calculates the probabilities that each input belongs to each of the available classes, the sum of all probabilities is equal to 1 [86,72]. Finally, the classification layer calculates the loss and assigns each spectrum to the class it has the highest probability of matching [87].

Internally, on each evaluation of a set of hyperparameters, stochastic gradient decent with momentum [71,72] is utilized. The goal is once again is to minimize loss, which is calculated in the classification layer. In figure 3.13 the starting point for stochastic gradient decent with momentum is the black dot. The gradient decent looks at points around the starting point and finds the gradient i.e., the steepest route, then we step in that direction as shown by the black arrow. The magnitude of the arrow is dictated by the value of the learning rate.

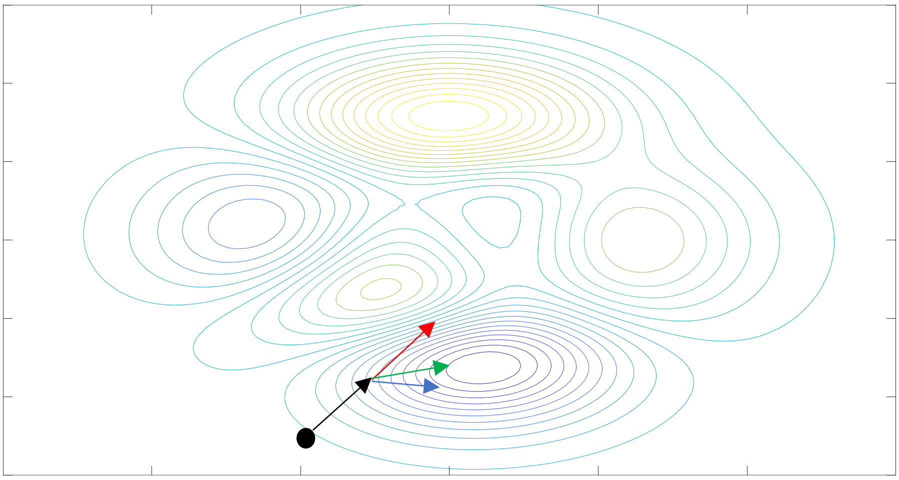


Figure 3.13: Analogy of stochastic gradient descent with momentum, starting point and first step in black, 2nd proposed step due to learning rate in blue, momentum in red and the actual second step in green [88].

If the learning rate is too large the minima cannot be approached and convergence will not be achieved. However, if the learning rate is too small the minima will not be approached within a reasonable time frame. A solution to this is to use the momentum approach.

In figure 3.13 the blue update vector shows the standard next step that would be taken towards the local minimum if momentum is not applied, it is simply an approach along the gradient. However, in the momentum approach the red update vector is calculated, an extension in the direction of the previous step. The green update vector then represents the next step to be taken, it is a weighted average of the blue and red update vectors. Momentum can be applied with a weighting (γ) of the blue and red update vectors ranging from 0 to 1. When its 0 the blue update vector dominates and becomes the resulting green update vector, when the weight is set to 1 the red update vector dominates the resulting green update vector.

In effect this allows us to approach the minimum in an almost curved like fashion without needing to lower the learning rate. The learning rate, η, dictates the size of the steps taken to reach a local minimum. In a model the updated parameters θ can be expressed in terms of their previous values, θ', the objective function, $J(\theta)$, the gradient, ∇_θ, and the learning rate as

$$\theta' = \theta - \eta \nabla_\theta J(\theta) \qquad (3.12).$$

$\nabla_\theta J(\theta)$ represents a decrease in height of the representational feature space as a local minimum is approached, hence the name stochastic gradient descent. The momentum approach can then be described by equation 3.13, where V_{t-1} is the red update vector, $\eta \nabla_\theta J(\theta)$ is the blue update vector, γ the weighting and V_t is the green update vector [88].

$$V_t = \gamma V_{t-1} + \eta \nabla_\theta J(\theta) \qquad (3.13).$$

3.3 Transformation of experimental spectra to inputs for CNNs

The largest issue to overcome in applying CNNs to LIBS spectra is turning the experimental spectra into an image so the CNNs will accept them as an input. The inspiration for a solution came from research on speech analysis using mel cepstrum [90-97], in which audio, a 1D signal like LIBS spectra, is converted into a 2D representation and presented to a CNN. The mel cepstrum is widely available tool commonly used in audio analysis. It is a representation of the short-term power spectrum of a sound over the mel(ody) frequencies, based on the inverse Fourier transform of the logarithm of the signal spectrum. The mel filter banks and frequency scaling are applied, which simulate human hearing, with good resolution at lower frequencies and lower resolution at higher frequencies. This filtering is not appropriate for our application. By stripping away all the unwanted filtering from the mel cepstrum approach to generating a 2D output, a short time Fourier transform (STFT) is what remains, and this will be the approach taken in this work.

The combination of STFT and CNN has gained significant traction in studies of electroencephalogram signals [98-103], signals from sensors for fault detection in the automotive industry [104,105] and the detection of unmanned aerial vehicles [106]. Equations 3.14 to 3.17 outline the STFT method. In equation 3.14 x is a real-world discrete signal, m is the local time index, l is the frame index, H is the hop size, $x[m + lH]$ is a segment of x and $W[m]$ is a finite window function. $x_l[m]$ is then the convolution of a window function operating on sliding segments of x [107]. The window function applied in this study is the Hamming window (equation 3.15, with N defined by equation 3.16 [108]).

$$x_l[m] = x[m + lH]W[m] \qquad (3.14),$$

$$W(n) = 0.54 - cos(2\pi \frac{n}{N}), \qquad 0 \leq n \leq N \qquad (3.15),$$

$$Window\ length = N + 1 \qquad (3.16).$$

The STFT of x, $\dot{X}[k, l]$, which is a discreet Fourier transform (DFT) performed on every frame of $x_l[m]$ is then given by:

$$\dot{X}[k,l] = \frac{1}{N}\sum_{m=1}^{K} x_l[m]e^{-i2\pi\frac{mk}{K}} \qquad (3.17),$$

where K corresponds to the DFT size and k corresponds to the frequency bin [107].

Figure 3.14 demonstrates the near 1 to 1 preservation of the spectra when converted into a 2D image, the final stage of down sampling preserving enough information to classify pharmaceutical LIBS spectra. It should be noted that while the work in this thesis is not the first in LIBS to use a CNN, it is the first to turn a single spectrum into a usable image for pharmaceutical classification in LIBS. Other efforts in LIBS have used a large volume of time resolved spectra and PCA to shape the resulting output into a single image [109], or a wavelength scan consisting of many spectra which was then reshaped into an image [110]. An advantage of the STFT+CNN approach is that only a single spectrum is required to generate an image. The approach taken in this thesis allows for more of the analyte to be probed and for individual spectra to be classified by a CNN.

Now that the recorded spectra are in a suitable image format all the components for the CNNs are complete. Once the experimental spectra are recorded, they undergo the STFT and the resulting images, along with the known class for each image, are passed to the network. The various convolution and down sampling layers produce the feature maps from these images. The fully connected layer, which connects all the higher layers, then transforms and combines all these feature maps into a model, and associated weights, for the system. This layer, in conjunction with the SoftMax layer, optimizes the hyperparameters to train the network to associate features and weights with a particular class. Once the network is fully trained, test images (STFT of experimental spectra) are passed to the network. The same layers operate on these test spectra and the resulting feature maps are used to predict which class a test image belongs to. If this predicted class corresponds to the actual known class, then the network is successful [41-49].

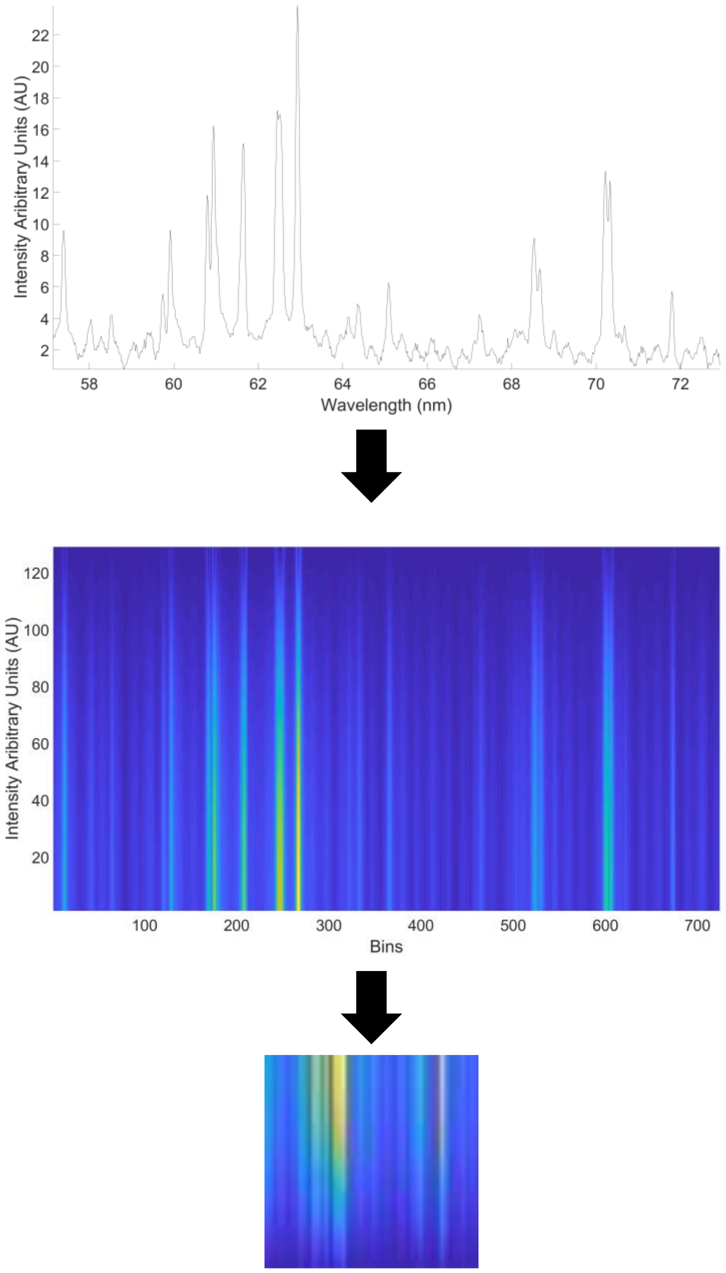


Figure 3.14: Conversion of sample spectrum to spectrogram then to 50x50 pixel image.

3.4 Machine learning algorithms

In this section the machine learning algorithms / frameworks used to classify samples will be described, from being presented with data to producing results. The following proprietary MATLAB functions were used to implement the machine learning algorithms. The competlayer function was used to carry out machine learning with competitive layers and define the neural network architecture [111]; the selforgmap function is used to carry out machine learning with SOMs and define the neural network architecture [112]; and the fitcsvm function was used to carry out machine learning with a SVM and define its architecture [113], selecting a radial basis function (RBF) or linear kernel.

A linear kernel is suited to applications where data is linearly separable, i.e., when a straight-line decision boundary can be drawn to separate classes. In an instance where that is not possible it may become necessary to employ an RBF kernel that can generate curved decision boundaries. A RBF kernel does have its drawbacks, while it can attain a nonlinear fit, in order to do so it has more parameters to compute thereby making it far more computationally expensive [114,115].

Finally, I used the deep network designer, a proprietary MathWorks tool [116] to design the architecture of the 7-layer CNN in section 3.2.4.

An overarching framework was developed to facilitate machine learning classification in this thesis. Code seen in the appendix C details the exact implementation for each type of machine learning. The framework serves two purposes, to present the machine learning technique with the training and testing data, and to save the results of the classification. This framework had to be mailable enough to be applied with little to no changes to facilitate each machine learning technique.

Standard across all studies was the following pre-processing of spectra: Each spectrum was normalised to the largest value within itself. This is done to minimise shot to shot variance.

COMP and SOMs use the normalised spectra as inputs. SVMs require inputs that are vectors of length 2 (two numbers long), so the normalised spectra are dimensionally reduced to their first two PC scores. Those PC scores are presented as inputs for the SVMs. This highlights the utility of PCA as its ability to implement dimensional reduction allowed SVMs to be added to the suite of machine learning techniques evaluated in this thesis. CNNs begin with normalised spectra, that then undergo the STFT process outlined in section 3.3 to produce images for classification. No other pre-processing of the data takes place.

3.4.1 Competitive learning

Implementing a COMP algorithm comes with facilitating an understanding of this algorithms shortcomings, COMP as described in section 3.2.1 is a winner take all clustering mechanism. When comparing this to SOMs, as described in section 3.2.2, COMP activates all its neurons for all inputs and competition takes place between all neurons until one wins. This is a computationally inefficient process compared to a decaying update radius in a SOM, which activates fewer and fewer neurons as inputs cluster over iterations. When more neurons are used in COMP the run times increase dramatically, greatly constraining the number of arrangements that can be evaluated in this study.

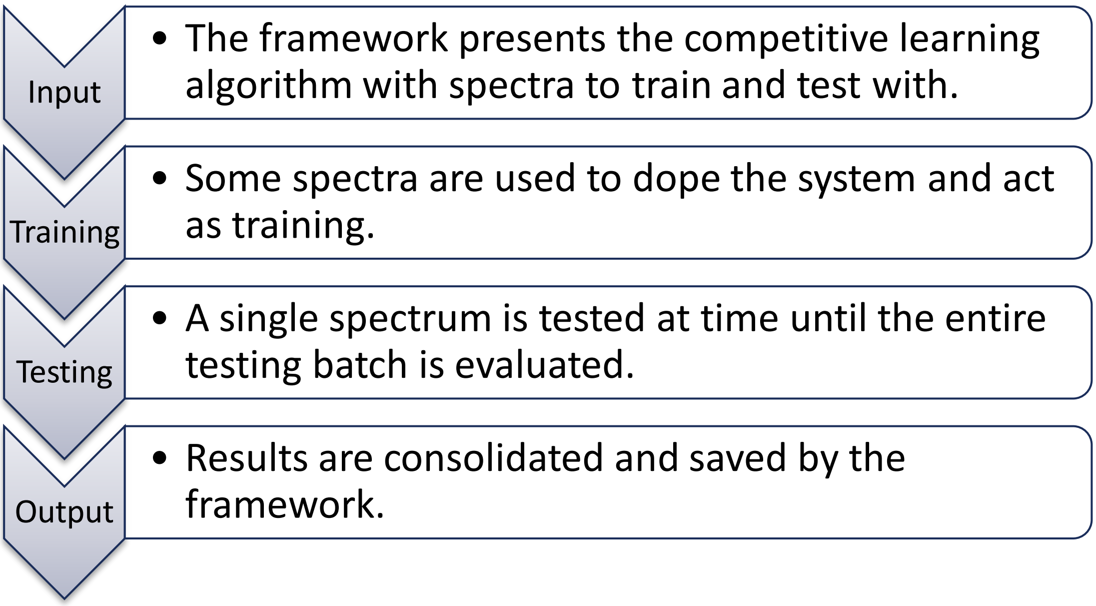
In order to evaluate COMP within a reasonable time frame only two arrangements were utilized. An arrangement with 2 neurons for binary testing and an arrangement with 4 neurons for multiclass testing. For COMP the framework presents spectra to the COMP algorithm to train and test with.

The methodology for training and testing as facilitated by the framework, and the associated conditional logic for classification is described as follows: Some spectra are used to dope the system and act as “training”, with a single spectrum tested each time. The training spectra and testing spectrum are presented to the network and register to neurons, this is also known as hitting them. The network has no knowledge of what class each presented spectrum belongs to.

The training spectra from each class will cluster on neurons such that only spectra belonging to that class will populate those neurons, i.e., each class will classify on a single neuron. The actual class each test spectrum belongs to is known by the framework. If a test spectrum hits a neuron it has a high probability of belonging to the class associated with that neuron.

Classification is successful if two conditions are met. Firstly, that a test spectrum hits a neuron belonging to the correct class. Secondly, the neuron hit does not belong to any other class. If the test spectrum hits a neuron belonging to the incorrect class or if the test spectrum does not hit a neuron belonging to its actual class classification is unsuccessful.

The results of this testing are collected as each test spectrum is evaluated. The results are consolidated by the framework. Those results are saved once all 5 runs, each with random assignments of spectra to training and testing batches as described in section 3.2, are evaluated. Figure 3.15 is a flowchart of the competitive neural networks algorithm.



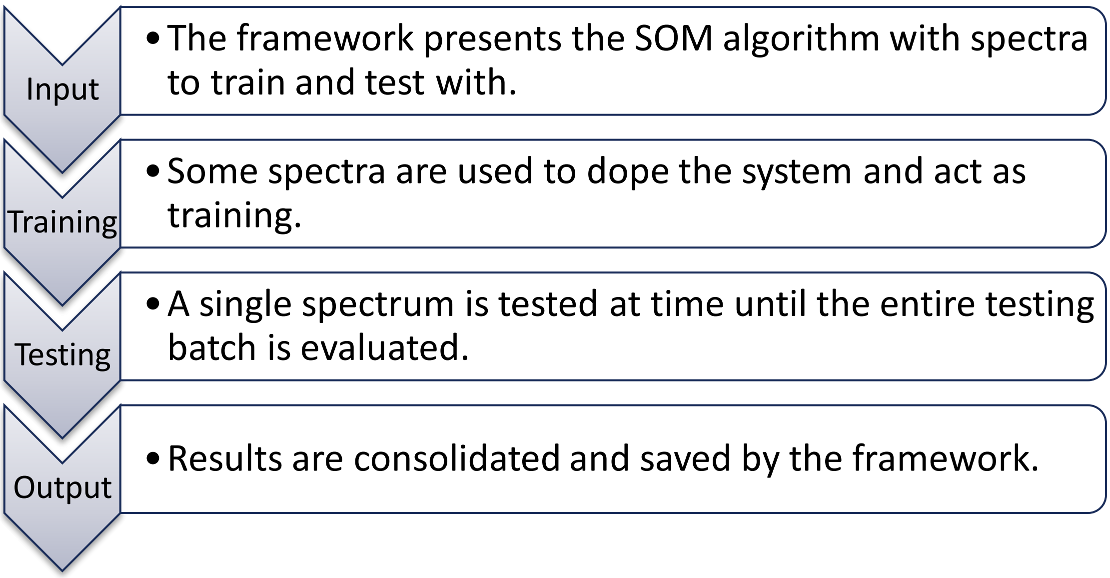
- The framework presents the competitive learning algorithm with spectra to train and test with.
- Some spectra are used to dope the system and act as training.
- A single spectrum is tested at time until the entire testing batch is evaluated.
- Results are consolidated and saved by the framework.

Figure 3.15: Flowchart of the COMP algorithm/framework detailing the sequence of processes.

3.4.2 Self-organizing maps

Several arrangements, each with a different number of neurons, were evaluated. The nature of each sub-layout will be identified as SOM axb, where a and b are the number of neurons initialized in each direction (e.g., SOM 5x5 consists of 25 neurons organized in a 5 by 5 grid). The methodology for training and testing, as facilitated by the framework, and the associated conditional logic for classification for SOMs is the same as described in section 3.4.1 for COMP with the exception that now a class may classify on many neurons, but those neurons must be unique from other classes.

Figure 3.16 is a flowchart of the SOM algorithm/framework. As with the other techniques 5 runs were conducted, each with random assignment of spectra to the training and testing batches. The results of all 5 runs are consolidated and saved by the framework.



- The framework presents the SOM algorithm with spectra to train and test with.
- Some spectra are used to dope the system and act as training.
- A single spectrum is tested at time until the entire testing batch is evaluated.
- Results are consolidated and saved by the framework.

Figure 3.16: Flowchart of self-organizing maps algorithm detailing the sequence of processes.

3.4.3 Support vector machines

The framework presents randomly assigned input spectra for training and testing. This random assignment was again carried out 5 times and 5 runs were evaluated; this was done to minimize the effect of a single spectrum being assigned to a particular batch. Spectra were initially dimensionally reduced using PCA, so each spectrum was subsequently represented by 2 principal components.

Bayesian optimization finds the optimal hyperparameters for the SVM by varying them and carrying out k-fold cross validation on the SVMs at those values of the hyperparameters. The training dataset is used to validate i.e., it is broken up into training and testing batches for the k-fold cross validation as illustrated in figure 3.17.

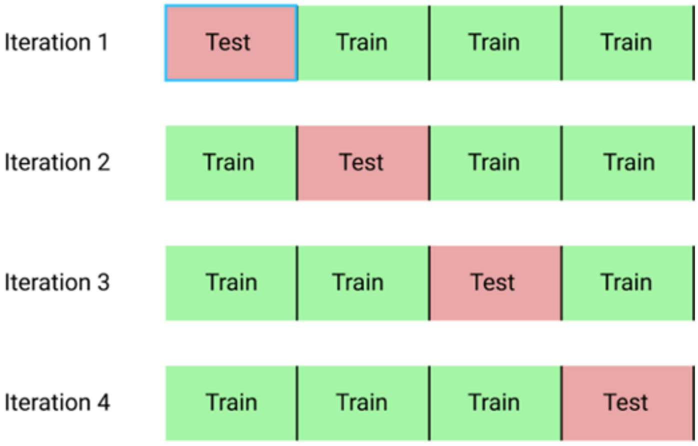


Figure 3.17: Illustration of k-fold cross validation where k=4 [117].

The variable k refers to the number of equally populated groups the data is divided into. One group is set aside for testing each time, and the others are used for training. All groups are cycled through until each one has acted as a test set. If k is set to the number of total number of spectra prior to the groups being divided, that would be leave one out cross validation, as each spectrum would act as a test set once. The trained network is used to carry out predictions on the testing dataset, if the predicted class is the same as the actual class then classification is successful. Results are consolidated by the framework and finally saved once all 5 runs are completed. Figure 3.18 is a flowchart of the framework and SVM algorithm.

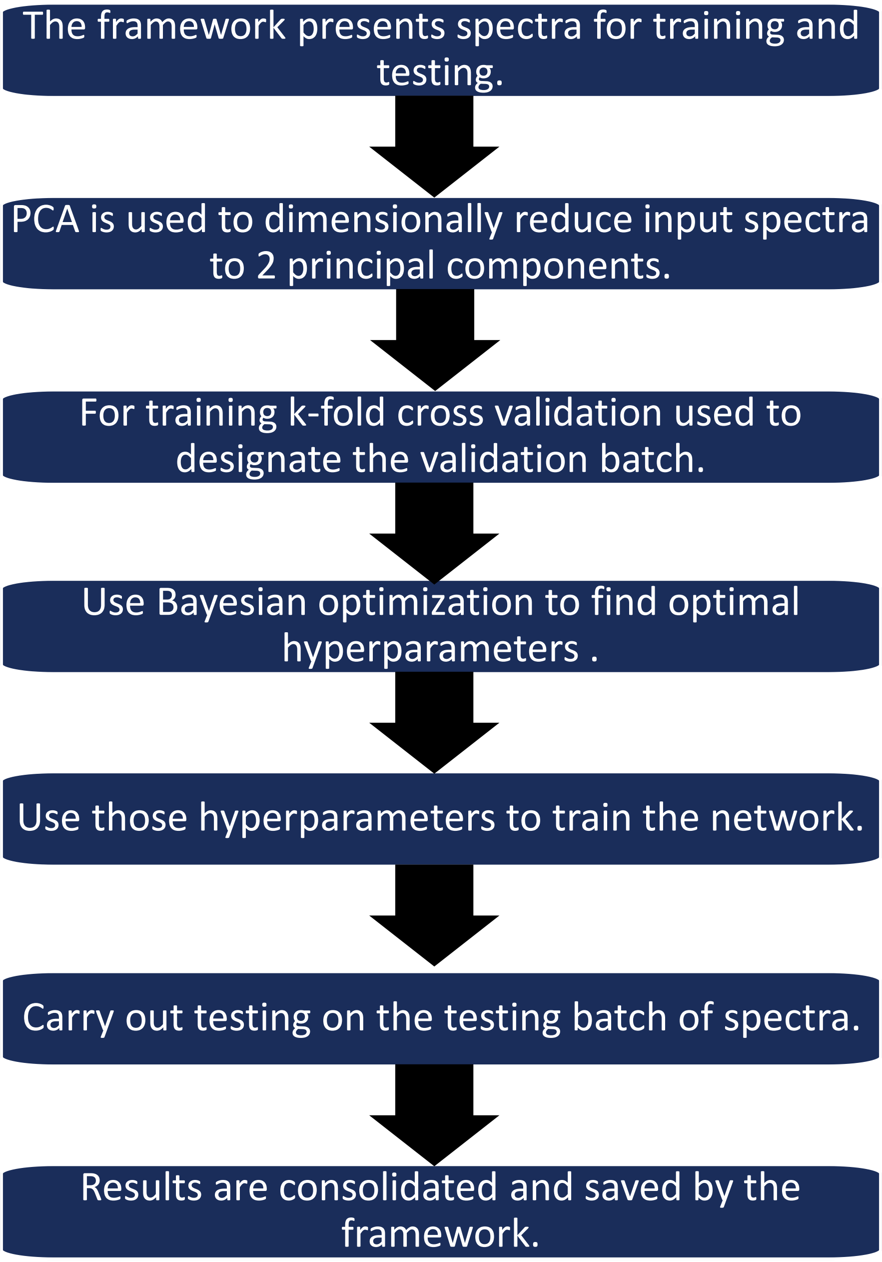


Figure 3.18: Flowchart of SVM algorithm detailing the sequence of processes.

3.4.4 Convolutional neural networks

For CNNs the framework presents spectra to the algorithm. The respective spectra are operated on with the STFT, then resized into 50 by 50 matrices, before a colour map is applied and they are saved as JPGs in file paths that denote their class labels. The images corresponding to spectra are divided, again, into a training batch, a validation batch and a testing batch. During training the spectra in the training and validation batches are shuffled once. Bayesian optimization is used to find the optimal hyperparameters for the CNN and the networks are trained and validated using stochastic gradient decent with momentum.

Classification is carried out with the testing batch. If the predicted class matches the actual class provided by the label, then classification is successful. There are 5 runs, each with a random assignment of spectra to training, validation and testing batches. Results are saved by the framework once all 5 runs are completed. Figure 3.19 is a flowchart of the CNN algorithm.

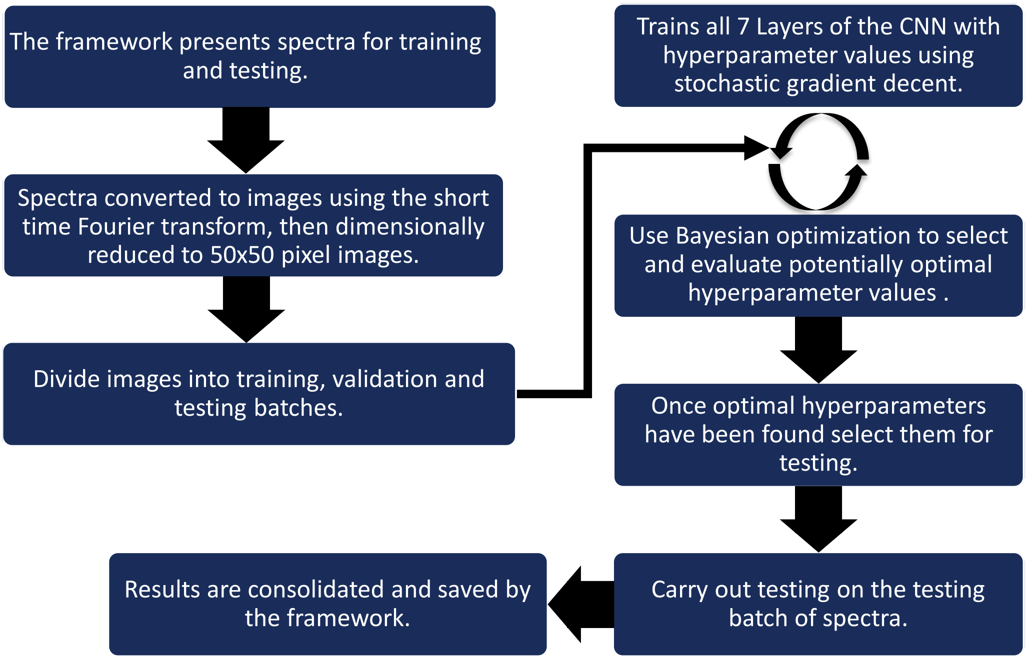


Figure 3.19: Flowchart of convolutional neural network algorithm detailing the sequence of processes.

3.5 Summary

In this chapter the general theory behind the machine learning techniques has been given, with COMP, SOMs, SVMs and CNNs described individually. The algorithms behind each technique applied in this thesis have also been outlined and the framework used to present experimental spectra to the machine learning algorithms, and collect the results of classification, has been described.

References

[1] Grossberg S. Competitive learning: From interactive activation to adaptive resonance. Cognitive science. 1987 Jan 1;11(1):23-63.

[2] Hofmann T, Buhmann JM. Competitive learning algorithms for robust vector quantization. IEEE Transactions on Signal Processing. 1998 Jun;46(6):1665-75.

[3] Ahalt SC, Chen P, Chou CT, Jung TP. Implementation of a vector quantization codebook design technique based on a competitive learning artificial neural network. The Journal of Supercomputing. 1992 Feb 1;5(4):307-30.

[4] McNeill DK, Card HC. Competitive learning and its application in adaptive vision for autonomous mobile robots. Connection Science. 1999 Dec 1;11(3-4):345-57.

[5] Bouzerdoum A. Image compression using a stochastic competitive learning algorithm (SCOLA). In Proceedings of the Sixth International Symposium on Signal Processing and its Applications (Cat. No. 01EX467) 2001 (Vol. 2, pp. 541-544). IEEE.

[6] Liu J, Liu ZQ. Self-splitting competitive learning for RBF network and speech data clustering. InProceedings. International Conference on Machine Learning and Cybernetics 2002 Nov 4 (Vol. 4, pp. 1858-1862). IEEE.

[7] Timouyas M, Hammouch A, Eddarouich S. A new approach of classification for non-Gaussian distribution upon competitive training. In IEEE International Conference on Complex Systems (ICCS) 2012 Nov 5 (pp. 1-6). IEEE.

[8] Khatter H, Aggarwal V, Ahlawat AK. Performance Analysis of the Competitive Learning Algorithms on Gaussian Data in Automatic Cluster Selection. In Second International Conference on Computational Intelligence & Communication Technology (CICT) 2016 Feb 12 (pp. 48-53). IEEE.

[9] Li T, Pei WJ, Wang SP, Cheung YM. Cooperation controlled competitive learning approach for data clustering. In International Conference on Computational Intelligence and Security 2008 Dec 13 (Vol. 1, pp. 24-29). IEEE.

[10] Kia SJ, Coghill GG. Unsupervised clustering and centroid estimation using dynamic competitive learning. Biological cybernetics. 1992 Sep 1;67(5):433-43.

[11] Card HC, Kamarsu S, McNeill DK. Competitive learning and vector quantization in digital VLSI systems. Neurocomputing. 1998 Jan 1;18(1-3):195-227.

[12] Choy CS, Siu WC. A class of competitive learning models which avoids neuron underutilization problem. IEEE transactions on neural networks. 1998 Nov;9(6):1258-69.

[13] Ma J, Wang T, Xu L. Convergence analysis of rival penalized competitive learning (RPCL) algorithm. In Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN'02 (Cat. No. 02CH37290) 2002 (Vol. 2, pp. 1596-1601). IEEE.

[14] Van Hulle MM, Martinez D. On a novel unsupervised competitive learning algorithm for scalar quantization. IEEE transactions on neural networks. 1994 May;5(3):498-501.

[15] Kosko B. Stochastic competitive learning. IEEE Transactions on Neural Networks. 1991 Sep;2(5):522-9.

[16] Zhang YJ, Liu ZQ. Self-splitting competitive learning: a new on-line clustering paradigm. IEEE Transactions on Neural Networks. 2002 Mar;13(2):369-80.

[17] Yair E, Zeger K, Gersho A. Competitive learning and soft competition for vector quantizer design. IEEE transactions on Signal Processing. 1992 Feb;40(2):294-309.

[18] Uchiyama T, Arbib MA. Color image segmentation using competitive learning. IEEE Transactions on pattern analysis and machine intelligence. 1994 Dec;16(12):1197-206.

[19] Samarasinghe S. Neural Networks for Applied Sciences and Engineering. Boca Raton, FL: Auerbach; 2007.

[20] Haykin S. Neural Networks A comprehensive foundation. Upper Saddle River, NJ: Prentice Hall; 1999.

[21] Ritter H, Martinetz T, Schulten K. Neural Computation and Self-Organizing Maps an Introduction Reading, Mass: Addison-Wesley; 1991.

[22] Pagnotta S, Grifoni E, Legnaioli S, Lezzerini M, Lorenzetti G, Palleschi V. Comparison of brass alloys composition by laser-induced breakdown spectroscopy and self-organizing maps. Spectrochimica Acta Part B: Atomic Spectroscopy. 2015 Jan 1;103:70-5.

[23] Chaplot S, Patnaik LM, Jagannathan NR. Classification of magnetic resonance brain images using wavelets as input to support vector machine and neural network. Elsevier Biomed Signal Process Control. 2006; 1:86–92.

[24] Arias C, Bani S, Catalli F, Lorenzetti G, Grifoni E, Legnaioli S, Pagnotta S, Palleschi V. X-ray fluorescence analysis and self-organizing maps classification of the Etruscan gold coin collection at the Monetiere of Florence. Applied spectroscopy. 2017 May;71(5):817-22.

[25] Hammer B, Villmann T. Neural maps in remote sensing image analysis ´. Elsevier Neural Networks. 2003; 16:389–403.

[26] Kohonen T. The self-organizing map. Elsevier Neurocomputing. 1998;21(May):1–6

[27] Kohonen T. Essentials of the self-organizing map. Elsevier Neural Networks. 2013; 37:52–65.

[28] Kangas J, Kohonen T. Developments and applications of the self-organizing map and related algorithms. Elsevier Math Comput Simul.1996;41:3–12.

[29] Klus J, Poˇ P. Application of self-organizing maps to the study of U-Zr-Ti-Nb distribution in sandstone-hosted uranium ores. Spectrochimica Acta Part B: Atomic Spectroscopy. 2017;131:66–73.

[30] Yu J, Hou Z, Sheta S, Dong J, Han W, Lu T, et al. Provenance classification of nephrite jades using multivariate LIBS: A comparative study. Anal Methods. 2018;10(3):281–9.

[31] Zhang C, Shen T, Liu F, He Y. Identification of Coffee Varieties Using Laser-Induced Breakdown Spectroscopy and Chemometrics. Sensors. 2017;18(1):95.

[32] Vahid Dastjerdi M, Mousavi SJ, Soltanolkotabi M, Nezarati Zadeh A. Identification and Sorting of PVC Polymer in Recycling Process by Laser-Induced Breakdown Spectroscopy (LIBS) Combined with Support Vector Machine (SVM) Model. Iran J Sci Technol Trans A Sci. 2018;42(2):959–65.

[33] Li X, Yang S, Fan R, Yu X, Chen D. Discrimination of soft tissues using laser-induced breakdown spectroscopy in combination with k nearest neighbors (kNN) and support vector machine (SVM) classifiers. Optics & Laser Technology. 2018 Jun 1;102:233-9.

[34] Yi C, Lv Y, Xiao H, Ke K, Yu X. A novel baseline correction method using convex optimization framework in laser-induced breakdown spectroscopy quantitative analysis. Spectrochimica Acta Part B: Atomic Spectroscopy. 2017 Dec 1; 138:72-80.

[35] Lu S, Dong M, Huang J, Li W, Lu J, Li J. Estimation of the aging grade of T91 steel by laser-induced breakdown spectroscopy coupled with support vector machines. Spectrochimica Acta Part B: Atomic Spectroscopy. 2018 Feb 1; 140:35-43.

[36] MathWorks.Support Vector Machines for Binary Classification [Internet].2017[Cited 13/02/1018]. Avialable from: https://www.mathworks.com/help/stats/support-vector-machines-for-binary-classification.html#bsr5o1q

[37] Hastie T, Tibshirani R, Friedman J. The elements of statistical learning: data mining, inference, and prediction. Springer Science & Business Media; 2009 Aug 26.

[38] Christianini, N, and J Shawe-Taylor. An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods. Cambridge, UK: Cambridge University Press, 2000.

[39] Fan RE, Chen PH, Lin CJ. Working set selection using second order information for training support vector machines. Journal of machine learning research. 2005;6(Dec):1889-918.

[40] Kecman V, Huang TM, Vogt M. Iterative single data algorithm for training kernel machines from huge data sets: Theory and performance. In Support vector machines: Theory and Applications 2005 Jun 21 (pp. 255-274). Springer, Berlin, Heidelberg.

[41] Dewa CK. Javanese vowels sound classification with convolutional neural network. Proceeding - 2016 Int Semin Intell Technol Its Appl ISITIA 2016 Recent Trends Intell Comput Technol Sustain Energy. 2017;123–8.

[42] Acar E, Hopfgartner F, Albayrak S. Fusion of learned multi-modal representations and dense trajectories for emotional analysis in videos. Proc - Int Work Content-Based Multimed Index. 2015;2015–July.

[43] Gajhede N, Beck O, Purwins H. Convolutional neural networks with batch normalization for classifying hi-hat, snare, and bass percussion sound samples. In Proceedings of the Audio Mostly 2016 2016 Oct 4 (pp. 111-115).

[44] Wang W, Plumbley MD. Convolutional Gated Recurrent Neural Network Incorporating Spatial Features for Audio Tagging. IJCNN. 2017;3461–6.

[45] Aykanat M, Kılıç Ö, Kurt B, Saryal S. Classification of lung sounds using convolutional neural networks. Eurasip J Image Video Process. 2017;2017(1).

[46] Lavner Y, Cohen R, Ruinskiy D, Ijzerman H, Database A. Baby Cry Detection in Domestic Environment using Deep Learning. ICSEE. 2016;

[47] Noda K, Yamaguchi Y, Nakadai K, Okuno HG, Ogata T. Audio-visual speech recognition using deep learning. Appl Intell. 2015;42(4):722–37.

[48] Chen B, Luo W, Li H. Audio steganalysis with convolutional neural network. In Proceedings of the 5th ACM Workshop on Information Hiding and Multimedia Security 2017 Jun 20 (pp. 85-90).

[49] Valenti M, Squartini S, Diment A, Parascandolo G, Virtanen T. A convolutional neural network approach for acoustic scene classification. In2017 International Joint Conference on Neural Networks (IJCNN) 2017 May 14 (pp. 1547-1554). IEEE.

[50] Hinton G. Neural Networks for Machine Learning [Internet]. 2013 [Cited 05/08/2018]. Available from: http://www.cs.toronto.edu/~hinton/coursera_slides.html

[51] Salatas J. Competitive neural network architecture [Internet] 2012 [28/07/2019]. Available from: https://commons.wikimedia.org/wiki/File:Competitive_neural_network_architecture.png

[52] Mathworks. Cluster with a Competitive Neural Network [Internet]. 2017 [Cited 28/05/2019]. Available from: https://uk.mathworks.com/help/deeplearning/ug/cluster-with-a-competitive-neural-network.html

[53] McClelland JL, Rumelhart DE, PDP Research Group. Parallel distributed processing. Explorations in the Microstructure of Cognition. 1986; 2:216-71.

[54] Bouzerdoum A. A stochastic competitive learning algorithm. InIJCNN'01. International Joint Conference on Neural Networks. Proceedings (Cat. No. 01CH37222) 2001 (Vol. 2, pp. 908-913). IEEE.

[55] Yang TN, Wang SD. Competitive algorithms for the clustering of noisy data. Fuzzy Sets and Systems. 2004 Jan 16;141(2):281-99.

[56] Burke LI. Competitive learning based approaches to tool-wear identification. IEEE transactions on systems, man, and cybernetics. 1992 May;22(3):559-63.

[57] Card HC, Rosendahl GK, McNeill DK, McLeod RD. Competitive learning algorithms and neurocomputer architecture. IEEE Transactions on Computers. 1998 Aug;47(8):847-58.

[58] Carpenter GA, Grossberg S. A massively parallel architecture for a self-organizing neural pattern recognition machine. Computer vision, graphics, and image processing. 1987 Jan 1;37(1):54-115.

[59] Lazzaro J, Ryckebusch S, Mahowald MA, Mead CA. Winner-take-all networks of O (n) complexity. In Advances in neural information processing systems 1989 (pp. 703-711).

[60] Kohonen T, Honkela T. Kohonen network. Scholarpedia. 2007 Jan 18;2(1):1568.

[61] Sutton GG, Reggia JA. Effects of normalization constraints on competitive learning. IEEE transactions on neural networks. 1994 May;5(3):502-4.

[62] Elanwar R. Introduction to Neural networks (undergraduate course) Lecture 9 of 9 [Internet]. 2015 [Cited 29/07/2019]. Available from: https://www.slideshare.net/Randa_Elanwar/introduction-to-neural-networks-under-graduate-course-lecture-9-of-9?from_action=save

[63] Mathworks. Cluster with Self-Organizing Map Neural Network [internet]. 2016 [Cited 21/07/2019]. Available from:

https://uk.mathworks.com/help/deeplearning/ug/cluster-with-self-organizing-map-neural-network.html

[64] Mathworks. Repeated copies of arrays [internet]. 2006 [Cited 05/09/2020]. Available from: https://uk.mathworks.com/help/matlab/ref/repmat.html

[65] Mathworks. Hexagonal layer topology function [internet]. 1997 [Cited 05/09/2020]. Available from: https://uk.mathworks.com/help/deeplearning/ref/hextop.html

[66] Datta B. Numerical methods for linear control systems. Academic Press; 2004.

[67] Banerjee A, Ghosh J. Frequency-sensitive competitive learning for scalable balanced clustering on high-dimensional hyperspheres. IEEE Transactions on Neural Networks. 2004 May;15(3):702-19.

[68] Ma J, Wang T. A cost-function approach to rival penalized competitive learning (RPCL). IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics). 2006 Aug;36(4):722-37.

[69] Landman N, Pandg H, Williams C, Ross E. k-Means Clustering [Internet]. 2018 [Cited 29/08/2018]. Available from: https://brilliant.org/wiki/k-means-clustering/

[70] Padmanabha A, Williams C. K-nearest Neighbors [Internet] 2018 [Cited 29/08/2018]. Available from: https://brilliant.org/wiki/k-nearest-neighbors/

[71] Bambrick N. Support Vector Machines: A Simple Explanation [internet]. 2016 [Cited 25/07/2019]. Available from: https://www.kdnuggets.com/2016/07/support-vector-machines-simple-explanation.html

[72] Bishop CM. Pattern Recognition and Machine Learning. Vol. 53, Journal of Chemical Information and Modeling. 2013. 1689-1699 p.

[73] Murphy KP. Machine learning: a probabilistic perspective. MIT press; 2012 Sep 7.

[74] Mockus J, Tiesis V, Zilinskas A. The application of Bayesian methods for seeking the extremum. Towards global optimization. 1978 Dec;2(117-129):2.

[75] Jones DR. A taxonomy of global optimization methods based on response surfaces. Journal of global optimization. 2001 Dec;21(4):345-83.

[76] Bull AD. Convergence rates of efficient global optimization algorithms. Journal of Machine Learning Research. 2011 Oct 1;12(10).

[77] Huijskens T. Bayesian optimization with scikit-learn [Internet]. 2016 [Cited 28/04/2018]. Available from: https://thuijskens.github.io/2016/12/29/bayesian-optimisation/#fnref:1

[78] Huijskens T. Bayesian optimization with scikit-learn [Internet]. 2017 [Cited 28/04/2018]. Available from: https://www.youtube.com/watch?v=jtRPxRnOXnk&t=1428s

[79] Becker D. Analysis of the Histogram Intersection Kernel for use in Bayesian Optimization. International Journal of Modeling and Optimization. 2017 Dec;7(6).

[80] Mathworks. Image input layer [Internet]. 2019 [Cited 28/07/2019] Available from: https://uk.mathworks.com/help/deeplearning/ref/nnet.cnn.layer.imageinputlayer.html

[81] Mathworks. 2-D convolutional layer [Internet]. 2019 [Cited 28/07/2019] Available from: https://uk.mathworks.com/help/deeplearning/ref/nnet.cnn.layer.convolution2dlayer.html

[82] Mathworks. Rectified Linear Unit (ReLU) layer [Internet]. 2019 [Cited 28/07/2019] Available from: https://uk.mathworks.com/help/deeplearning/ref/nnet.cnn.layer.relulayer.html

[83] Mathworks. Max pooling layer [Internet]. 2019 [Cited 28/07/2019] Available from: https://uk.mathworks.com/help/deeplearning/ref/nnet.cnn.layer.maxpooling2dlayer.html

[84] Computersciencewiki. Max-pooling / Pooling [Internet]. 2018 [Cited 28/07/2019] Available from: https://computersciencewiki.org/index.php/Max-pooling_/_Pooling

[85] Mathworks. Fully connected layer [Internet]. 2019 [Cited 28/07/2019] Available from: https://uk.mathworks.com/help/deeplearning/ref/nnet.cnn.layer.fullyconnectedlayer.html

[86] Mathworks. Softmax layer [Internet]. 2019 [Cited 28/07/2019] Available from: https://uk.mathworks.com/help/deeplearning/ref/nnet.cnn.layer.softmaxlayer.html

[87] Mathworks. Classification output layer [Internet]. 2019 [Cited 28/07/2019] Available from: https://uk.mathworks.com/help/deeplearning/ref/classificationlayer.html

[88] Mathworks. Deep Learning Onramp [Internet]. 2017 [Cited 28/04/2018]. Available from: https://www.mathworks.com/training-schedule/deep-learning-onramp

[89] Sutton R. Two problems with back propagation and other steepest descent learning procedures for networks. In Proceedings of the Eighth Annual Conference of the Cognitive Science Society, 1986 (pp. 823-832).

[90] Hyder R, Ghaffarzadegan S, Feng Z, Hansen JH, Hasan T. Acoustic Scene Classification Using a CNN-SuperVector System Trained with Auditory and Spectrogram Image Features. In INTERSPEECH 2017 Aug (pp. 3073-3077).

[91] Bae SH, Choi I, Kim NS. Acoustic scene classification using parallel combination of LSTM and CNN. In Proceedings of the Detection and Classification of Acoustic Scenes and Events 2016 Workshop (DCASE2016) 2016 Sep (pp. 11-15).

[92] Valenti M, Diment A, Parascandolo G, Squartini S, Virtanen T. DCASE 2016 acoustic scene classification using convolutional neural networks. InProc. Workshop Detection Classif. Acoust. Scenes Events 2016 Sep (pp. 95-99).

[93] Amoh J, Odame K. DeepCough: A deep convolutional neural network in a wearable cough detection system. In IEEE Biomedical Circuits and Systems Conference (BioCAS) 2015 Oct 22 (pp. 1-4). IEEE.

[94] Cai D, Ni Z, Liu W, Cai W, Li G, Li M, Cai D, Ni Z, Liu W, Cai W. End-to-End Deep Learning Framework for Speech Paralinguistics Detection Based on Perception Aware Spectrum. In INTERSPEECH 2017 (pp. 3452-3456).

[95] Ebbers J, Nelus A, Martin R, Haeb-Umbach R. Evaluation of modulation-MFCC features and DNN classification for acoustic event detection. Proc. of Jahrestagung für Akustik. 2018 Mar.

[96] Hoshen Y, Weiss RJ, Wilson KW. Speech acoustic modeling from raw multichannel waveforms. In IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) 2015 Apr 19 (pp. 4624-4628). IEEE.

[97] Takaki S, Nakashika T, Wang X, Yamagishi J. STFT spectral loss for training a neural speech waveform model. In ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) 2019 Apr 16 (pp. 7065-7069). IEEE.

[98] Shahbazi M, Aghajan H. A generalizable model for seizure prediction based on deep learning using CNN-LSTM architecture. In IEEE Global Conference on Signal and Information Processing (GlobalSIP) 2018 Nov 26 (pp. 469-473). IEEE.

[99] Choi G, Park C, Kim J, Cho K, Kim TJ, Bae H, Min K, Jung KY, Chong J. A Novel Multi-scale 3D CNN with Deep Neural Network for Epileptic Seizure Detection. In IEEE International Conference on Consumer Electronics (ICCE) 2019 Jan 11 (pp. 1-2). IEEE.

[100] Fangmeng Z, Peijia L, Iwamoto M, Kuwahara N. Emotional Changes Detection for Dementia People with Spectrograms from Physiological Signals. International Journal of Advanced Computer Science and Applications. 2018 Oct 1;9(10):49-54.

[101] Cao Y, Guo Y, Yu H, Yu X. Epileptic seizure auto-detection using deep learning method. In 4th International Conference on Systems and Informatics (ICSAI) 2017 Nov 11 (pp. 1076-1081). IEEE.

[102] Zhang J, Yan C, Gong X. Deep convolutional neural network for decoding motor imagery based brain computer interface. In IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC) 2017 Oct 22 (pp. 1-5). IEEE.

[103] Wang Z, Cao L, Zhang Z, Gong X, Sun Y, Wang H. Short time Fourier transformation and deep neural networks for motor imagery brain computer interface recognition. Concurrency and Computation: Practice and Experience. 2018 Dec 10;30(23):e4413.

[104] Liao Y, Zeng X, Li W. Wavelet transform based convolutional neural network for gearbox fault classification. In Prognostics and System Health Management Conference (PHM-Harbin) 2017 Jul 9 (pp. 1-6). IEEE.

[105] Wang LH, Zhao XP, Wu JX, Xie YY, Zhang YH. Motor fault diagnosis based on short-time Fourier transform and convolutional neural network. Chinese Journal of Mechanical Engineering. 2017 Nov 1;30(6):1357-68.

[106] Matson E, Yang B, Smith A, Dietz E, Gallagher J. UAV Detection System with Multiple Acoustic Nodes Using Machine Learning Models. In Third IEEE International Conference on Robotic Computing (IRC) 2019 Feb 25 (pp. 493-498). IEEE.

[107] Zhivomirov H. On the Development of STFT-analysis and ISTFT-synthesis Routines and their Practical Implementation. TEM Journal. 2019;8(1):56-64.

[108] Mathworks. Hamming [Internet]. 2019 [Cited 13/06/2019]. Available from: http:// https://uk.mathworks.com/help/signal/ref/hamming.html

[109] Chengxu LU, Bo WA, Jiang X, Zhang J, Kang NI, Yanwei YU. Detection of K in soil using time-resolved laser-induced breakdown spectroscopy based on convolutional neural networks. Plasma Science and Technology. 2018 Dec 27;21(3):034014.

[110] Chen J, Pisonero J, Chen S, Wang X, Fan Q, Duan Y. Convolutional neural network as a novel classification approach for laser-induced breakdown spectroscopy applications in lithological recognition. Spectrochimica Acta Part B: Atomic Spectroscopy. 2020 Apr 1;166:105801

[111] MathWorks. Competitive layer [Internet]. 2019 [Cited 09/07/2020]. Available from: https://uk.mathworks.com/help/deeplearning/ref/competlayer.html

[112] MathWorks. Cluster with Self-Organizing Map Neural Network [Internet].2016[Cited 09/07/2020]. Available from: https://uk.mathworks.com/help/deeplearning/ug/cluster-with-self-organizing-map-neural-network.html

[113] MathWorks. Fitcsvm [Internet]. 2017.[Cited 09/07/2020]. Available from: https://uk.mathworks.com/help/stats/fitcsvm.html

[114] Raschka S. How to Select Support Vector Machine Kernels. 2016. [Cited 21/06/2020]. Available from: https://www.kdnuggets.com/2016/06/select-support-vector-machine-kernels.html

[115] Schwenker F, Kestler HA, Palm G. Unsupervised and supervised learning in radial-basis-function networks. InSelf-Organizing neural networks 2002 (pp. 217-243). Physica, Heidelberg.

[116] Mathworks. Build Networks with Deep Network Designer [Internet]. 2019.[Cited 09/07/2020]. Available from: https://uk.mathworks.com/help/deeplearning/ug/build-networks-with-deep-network-designer.html

[117] Shaikh R. Cross Validation Explained: Evaluating estimator performance. [internet]. 2018 [Cited 25/07/2019] Available from: https://towardsdatascience.com/cross-validation-explained-evaluating-estimator-performance-e51e5430ff85

4 Initial classification of pharmaceuticals using VUV-LIBS and machine learning

This chapter contains the initial experimental results and a discussion on the performance of vacuum ultraviolet laser induced breakdown spectroscopy (VUV-LIBS) for pharmaceutical classification. As previously stated in chapter 1, I adapted the experimental system and acquired all spectra presented in this thesis. This initial study was performed under experimental conditions that are often considered advantageous in LIBS, i.e strong emission lines with a high signal to background ratio. The VUV-LIBS spectra are then combined with PCA and the machine learning approaches outlined in chapter 3 to determine classification capability of each technique. This was an important study as it provided a comparison to see how a poorly chosen and a well-chosen wavelength region would affect the machine learning classification and also highlighted some issues which require attention, such as inter shot spacing, excessive sample surface damage and the limitations due to sample curvature, which were used to inform subsequent studies.

The experimental spectra gathered for this study are presented in section 4.2.1. The spectra produced by this study had, in general, a high signal to noise ratio (as shown later in section 5.3.2), which in principle were spectra suitable for classification purposes. The method for the visualization of results using principal component analysis (PCA) is presented in section 1.6. The underlying theory behind the analytical techniques employed for the classification of analytes using machine learning is presented in section 3.2. Self-organizing maps (SOM), support vector machines (SVM) and convolutional neural networks (CNN) are the machine learning techniques used in this chapter. A detailed description of the machine learning algorithms is shown in section 3.4, while the code used to run those algorithms is shown in appendix C.

A subsequent and more comprehensive study, learning from, and adapting to, issues encountered in the initial study, was carried out and forms the basis of chapter 5.

4.1 VUV spectra from pharmaceutical samples

The basis of the VUV-LIBS system has been developed in Dublin City University for a number of years and has proven to be adaptable to a number of scientific investigations [1-8]. The details of the experimental apparatus used for this study are given in section 2.2.1. For all results presented in this chapter the vacuum in the chamber and spectrometer was maintained at a pressure of ~2 x 10^{-6} mbar. The 6 ns pulse from a Continuum Surelite laser was focused to a spot size of approximately 200 μm diameter on the sample, with a pulse energy of 460 mJ entering the target chamber. Four significantly differing tablets were used for this initial study: 2 pharmaceuticals, 1 nutraceutical and one food item.

It is often desirable to allow the plasma to expand and cool before collecting the radiation from it, to reduce self-absorption and continuum emission [9] as described in section 1.4.3. In this set of experiments the target surface was positioned 2 mm from the optical axis of the spectrometer to achieve this.

For a single grating position, a 17 nm spectral window can be obtained. Such spectra were recorded at 105 nm, 95 nm, 85 nm, 75 nm, 65 nm, 55nm and 45nm central wavelengths, giving spectral information from 37 to 112 nm. For convenience the different spectral regions will be referred to by their central wavelength (as a shorthand to ease the reading of this chapter). At each central wavelength 12 single shot time-integrated spatially resolved (TISR) spectra were recorded for each sample.

The time taken to measure the entire spectral range of each sample was approximately 30 minutes, however, as discussed in section 2.2, this was limited by the equipment used rather than by the principle of the technique. 12 shots per sample were chosen primarily due to delamination of the surface which limited how much of a single tablet could be probed, as outlined in section 2.2.2. Each of the 12 shots were recorded with the same focusing conditions and under the same experimental conditions, but at different sites on the sample. The 105 nm, 85 nm, 75 nm and 45 nm regions were investigated in greater detail. This was done so as to select two regions that were estimated to yield spectra that were distinctly different for the pharmaceutical samples, i.e., at 105 nm and 85 nm, while 45 nm and 75 nm were picked due to the similarity of the spectra between samples and the greater intra class variance within in those spectra.

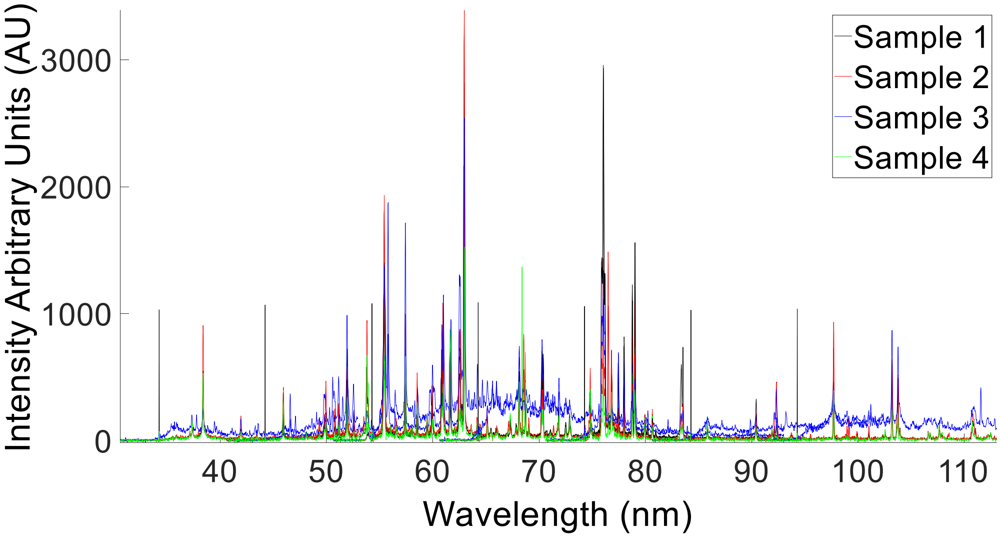


Figure 4.1: Mosaic spectrum over the entire spectral range of the spectrometer, for each of the samples used.

Mosaics of the spectra recorded for each are shown in figure 4.1. Where sample 1 is the multivitamin, sample 2 is the Disprin (containing aspirin), sample 3 is the Panadol (containing paracetamol) and sample 4 is the sweet with a similar visual and surface appearance to the other samples. In producing figure 4.1 the 12 spectra recorded for each sample per wavelength region are averaged. The spectra share several common lines which is unsurprising as they all contain carbohydrates and are of a similar elemental composition, as noted in their ingredients list in section 2.3. Some of these common spectral lines are identified in appendix D.

Looking at these averaged spectra some immediate observations can be made. In some regions there are very distinct differences, even to the eye, in the spectra. However, in other regions the differences are visible to a much lesser extent, with significant overlap of some samples while other samples are still quite dissimilar. This implies that the particular spectral region to be used for classification will be important, and there may not be an ideal spectral region for all purposes. This point is perhaps shown more clearly in figure 4.2, where the average spectrum for each sample in each spectral region is shown.

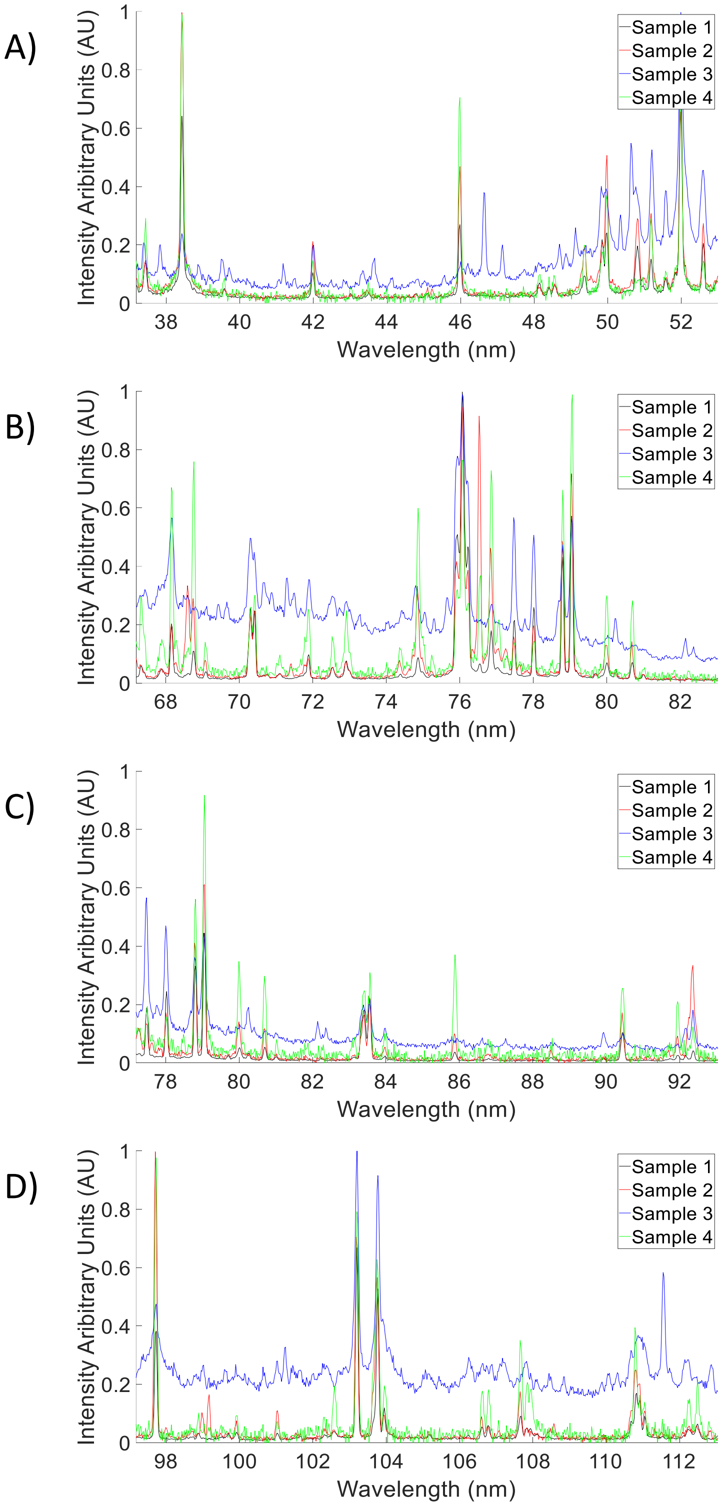


Figure 4.2: Average of 12 spectra for each sample at various central wavelengths. A) 45 nm central wavelength, B) 75 nm central wavelength, C) 85 nm central wavelength and D) 105 nm central wavelength.

When discussing the overlap of spectra within a class, a tighter overlap of spectra implies lower intra class variance. An equally important consideration is the variation of the individual single shot spectra within a given sample. If the variation of spectra from a sample is similar to, or indeed greater than, the variation of spectra between samples this would have an adverse effect on the classification potential of VUV-LIBS. A large number of spectra (and hence surface area) would be required for each sample to obtain a reliable averaged spectrum, limiting the processing time of samples and eliminating the possibility of single shot operation. In summary if the intra class variations are greater than the inter class variations the classification will be less reliable and, if averaging were used, a greater number of spectra would be required.

To visualise the intra class variations difference spectra for each sample were compared at various wavelength regions. The difference spectra were calculated as the difference between each normalised single shot spectrum relative to the normalised average spectrum for a class. The 12 single shot difference spectra for each of the samples recorded in a 17 nm bandwidth about 45 nm are shown in figure 4.3. While there is some variation in the intensity of the emission lines there is no obvious trend to these variations and, in general over the spectral region, there is a tight overlap of spectra for samples 1, 2 and 4, while there is less overlap of the spectra for sample 3. The tight overlap observed implies comparatively lower intra class variance and a resultant higher probability to detect inter class variations. In figure 4.3 sample 2 is an interesting case: the overlap is good, but a feature, just above 46 nm, appears in just one of the spectra for this sample. This shows the problem with using a simple line identification and/or a visual grouping of spectra as a classification method from LIBS spectra, especially with inhomogeneous, grainy, and non-cohesive structures.

In figures 4.4–4.6 similar single shot difference spectra, for each of the samples, are shown for regions with a central wavelength of 75 nm, 85 nm and 105 nm, respectively. Again, a similar trend is observed: some exhibit good overlap; some poor overlap; some show large variations in localized emission lines. The only obvious trend in figures 4.3–4.6 is the poor overlap in sample 3, implying there is much larger inter class variation.

It should be noted that the curvature of sample 3 was the greatest of all the tablets investigated. It would appear that, despite the care taken to ensure the same experimental conditions for each spectrum recorded, the curvature of this tablet had the most obvious impact on the spectra. This is not necessarily a bad thing as in this study the curvature of the tablet is considered a property of the tablet, so it is still valid to see if the techniques can classify based on it. Also, this is only present in one tablet so will not affect the results for the other classes.

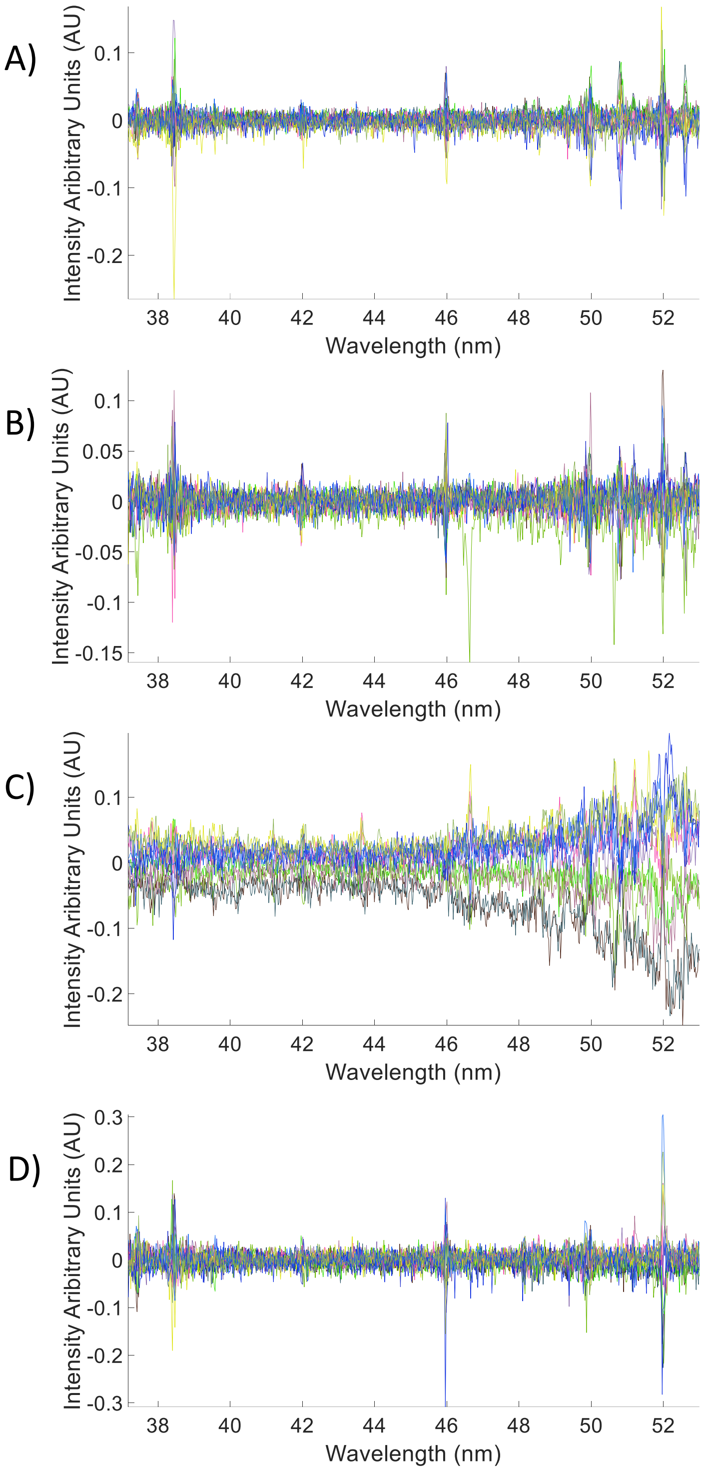


Figure 4.3: 12 single shot difference spectra with a central wavelength of 45 nm. A) sample 1, B) sample 2, C) sample 3 and D) sample 4.

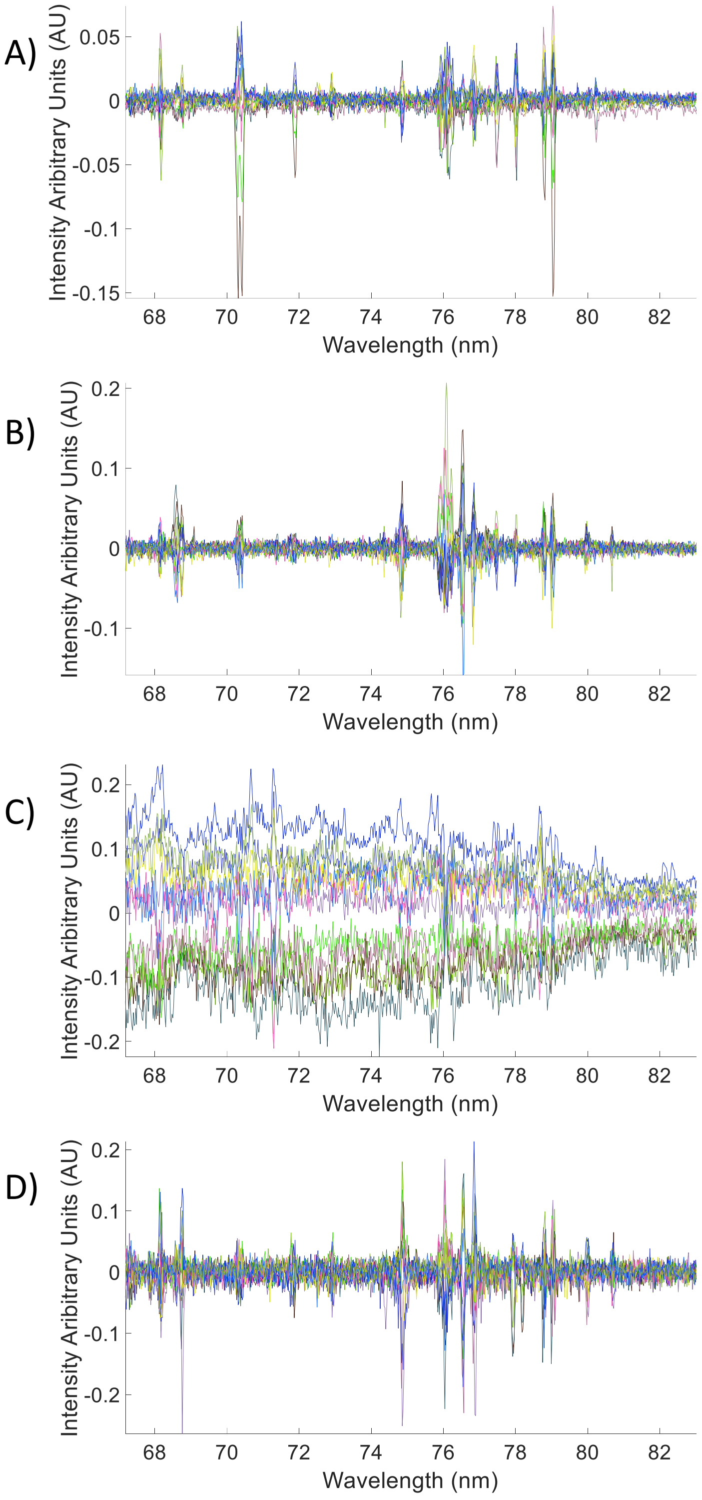


Figure 4.4: 12 single shot difference spectra with a central wavelength of 75 nm. A) sample 1, B) sample 2, C) sample 3 and D) sample 4.

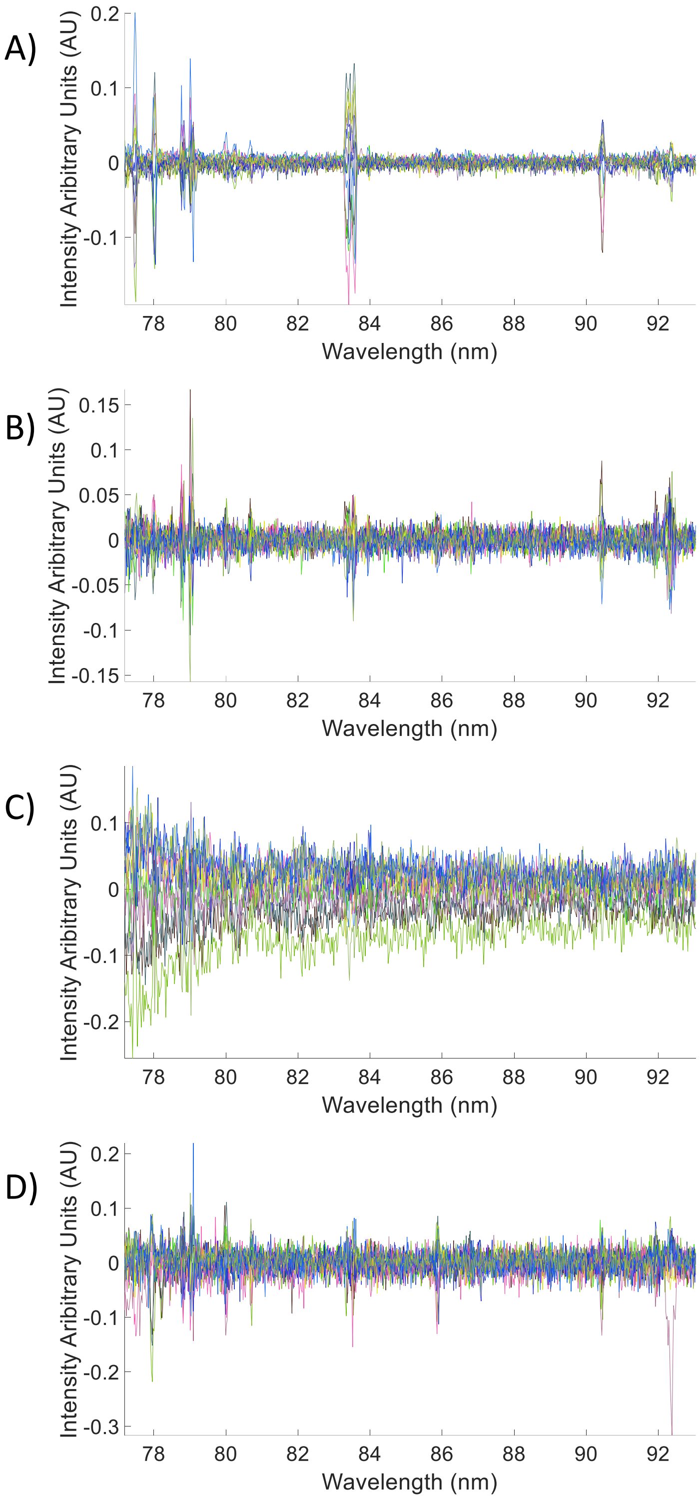


Figure 4.5: 12 single shot difference spectra with a central wavelength of 85 nm. A) sample 1, B) sample 2, C) sample 3 and D) sample 4.

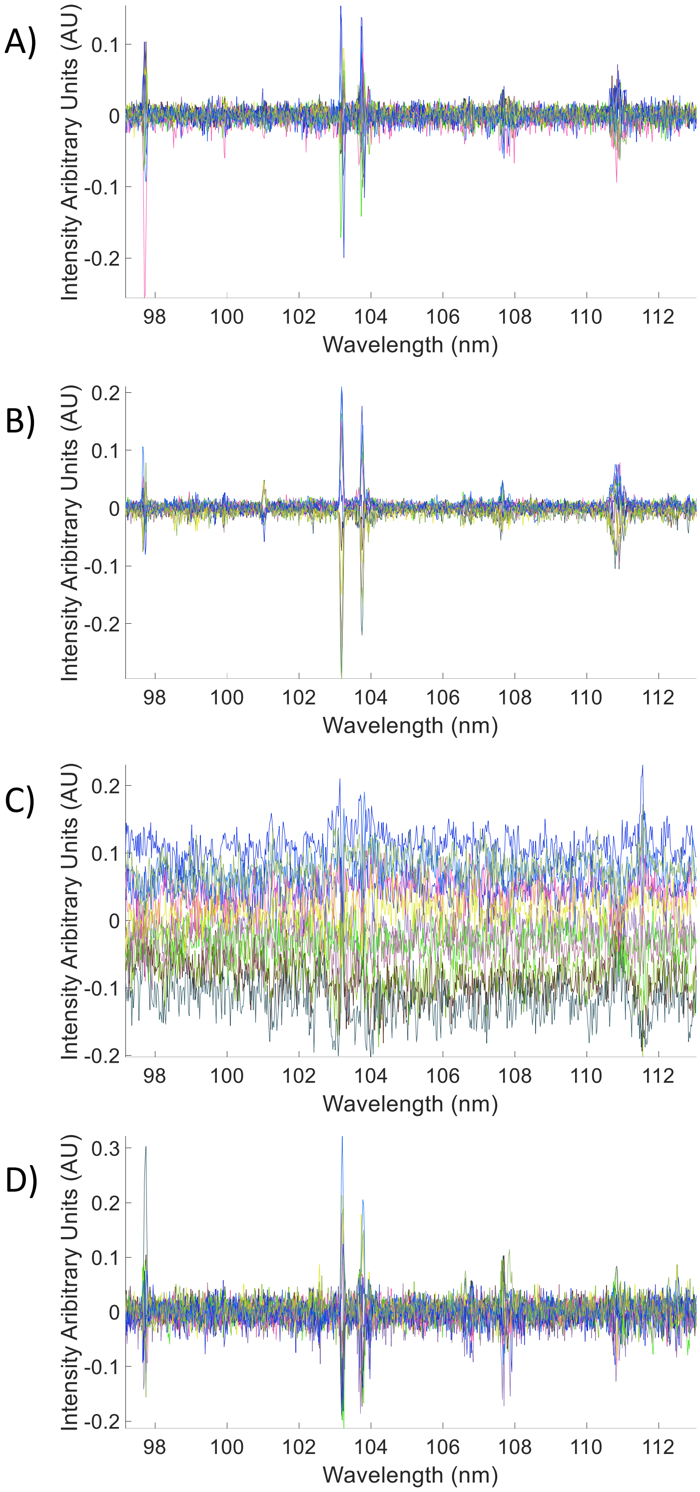


Figure 4.6: 12 single shot difference spectra with a central wavelength of 105 nm. A) sample 1, B) sample 2, C) sample 3 and D) sample 4.

4.2 Multivariate statistical analysis

The initial data set was analysed using the PCA [10] technique outlined in section 1.6. The aim of this analysis was two-fold. Firstly, to investigate if the recorded spectra could be used to distinguish between the different samples and secondly, to investigate the sensitivity of PCA. As established in section 1.5.2, the eigenvectors are sorted from highest to lowest eigenvalue. This has the effect of sorting them by most to least variance that they store for the entire population. The PC scores from these eigenvectors are then calculated and, here, the first three PCs are used as this preserves more variance than just the first two PCs. As an example, in the 105 nm central wavelength region, two PCs preserve 97.0% of the variance while three PCs preserve 98.1% of the variance. One could simply select and replace one of the chosen PCs with another from the list explicitly in the code shown in the Appendix C, but this would affect the variance retained.

4.2.1 Sample classification

For each sample all 48 spectra recorded for a given central wavelength region (12 for each sample) were used for the PCA. The first 3 principal component (PC) scores for each spectrum are used to represent the spectrum in a 3-D plot. Examples of these 3-D plots are shown in figure 4.7. As a reminder the ellipsoid classifiers (also shown in figure 4.7) are generated such that the centre points are the means of a data set in their respective directions and the semi axes correspond to three times the standard deviation of the PC score in a given direction.

From figure 4.7 there is successful classification, via separation of analytes into 4 distinct classes (i.e., there is no overlap in ellipsoid classifiers) in A) and B). However, there is a failure to classify in C) and D).

In the case of C) there is a larger intra class variance in class 3, which agrees with the observation from the difference spectra presented in figures 4.3, 4.4, 4.5 and 4.6. This large variance manifests as a large volume for the ellipsoid that defines the region of the 3-D space that is assigned to sample 3. This high spatial distribution of data points completely engulfs the other ellipsoids.

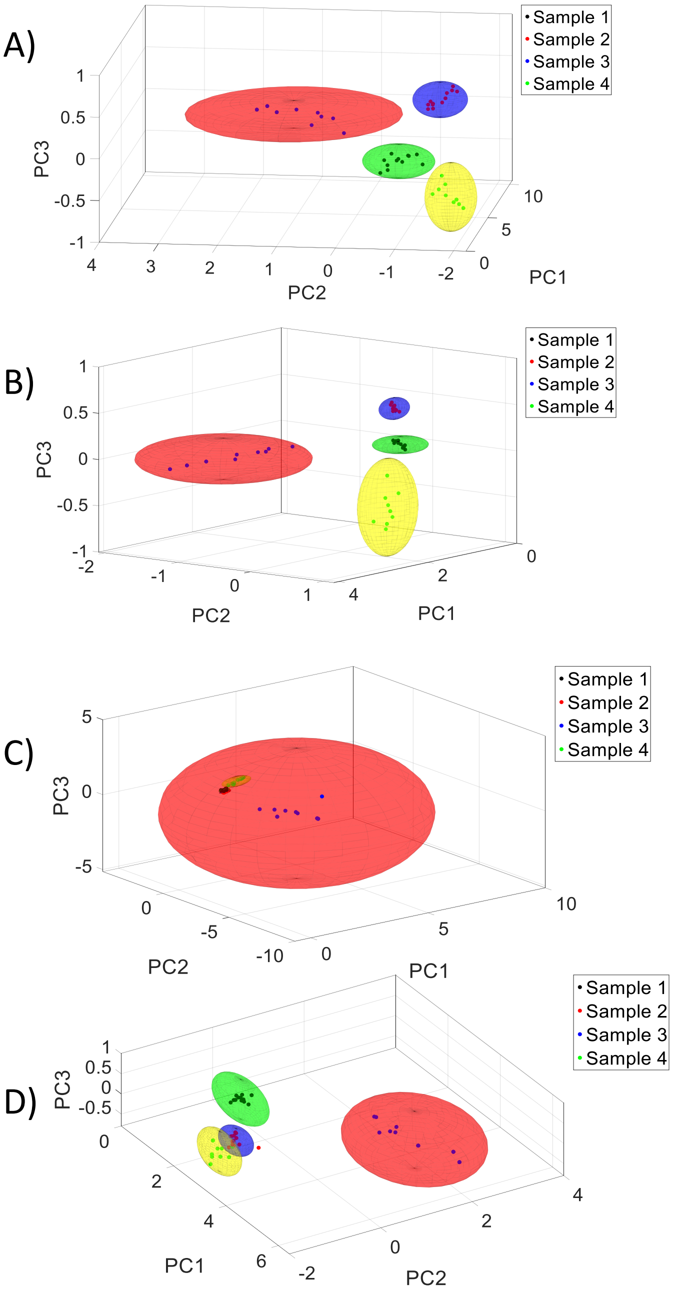


Figure 4.7: The first 3 PC scores for each spectrum, and corresponding ellipsoid classifiers, at various central wavelengths. A) 105 nm central wavelength, B) 85 nm central wavelength, C) 75 nm central wavelength, D) 45 nm central wavelength.

In the case of D) there is an overlap of the ellipsoids representing samples 2 and 4, so these cannot be distinguished from each other. From figure 4.2 A) the spectra for these two samples are similar in this wavelength region, so it is not surprising the similar spectra, when dimensionally reduced, generate points that appear close in PC space.

4.2.2 Sensitivity testing

As previously mentioned, the spectra recorded contained a property, which could hinder the classification of samples, namely the variation of localized emission lines. A test was designed to visualise this effect by artificially varying the intensity of an emission line in a recorded spectrum.

Implementation of this test is shown in figure 4.8. In figure 4.8 A) a synthetic line is added to a spectrum to simulate the variation of a single line, or indeed a limit of detection experiment. The relative intensity of the added Voigt profiles ranged from 0.1 to 1, as shown in 4.8 B).

A calibration curve typically has a spectral line intensity on the y axis and concentration of a trace element of interest of the analyte on the x axis. In some cases, the intensity parameter is in arbitrary units and referred to as relative intensity, the relative intensity being the integral of the area under a spectral line of an element of interest. That relative intensity is then related to the known concentration of a sample. A regression calculation takes place, and a calibration curve is drawn. For a relative intensity within the range of the calibration curve a corresponding concentration of the analyte can be estimated. This allows quantitative information regarding the concentration of an analyte to be extracted experimentally [11-14].

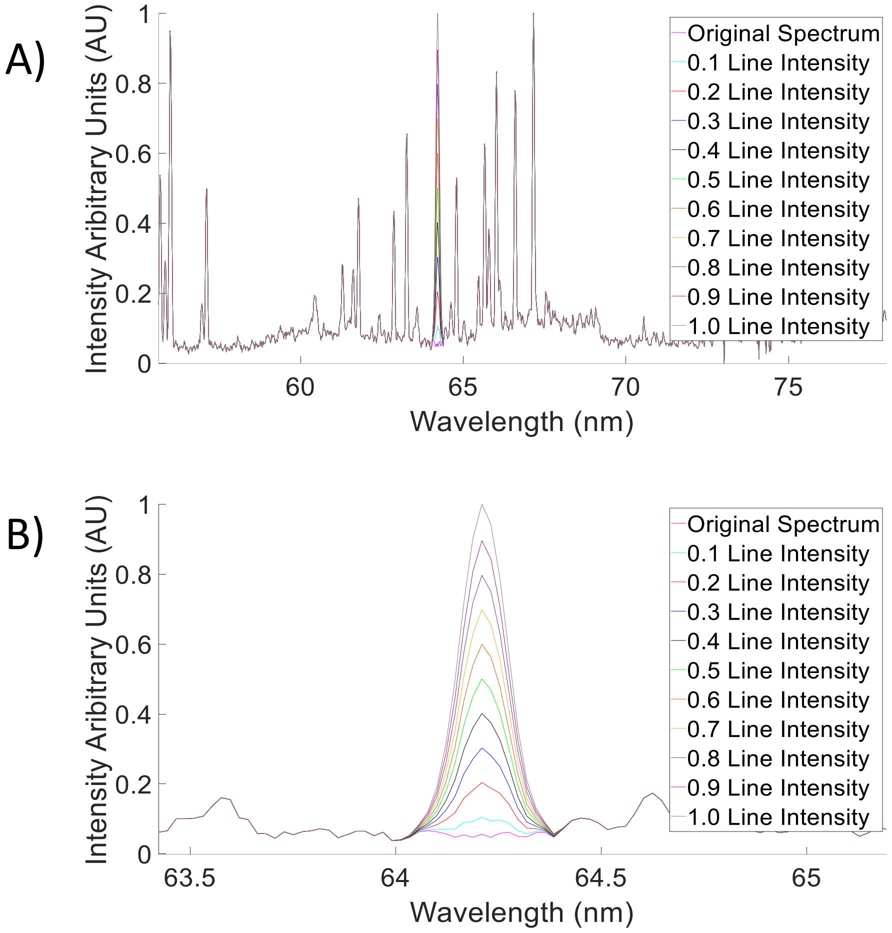


Figure 4.8: A) Voigt profile line added to an experimental spectrum and B) zoomed view of the Voigt line.

Here, PCA was again performed for the original spectrum and the 10 synthetic spectra, with the results shown in figure 4.9. As the intensity of the Voigt profile is increased linearly a view of the 3-D plot can be obtained that also shows a linear deflection as shown in figure 4.9 A), drawing comparison to the behaviour of a calibration curve as shown in figure 4.9 B) traditionally used in quantitative LIBS analysis [15].

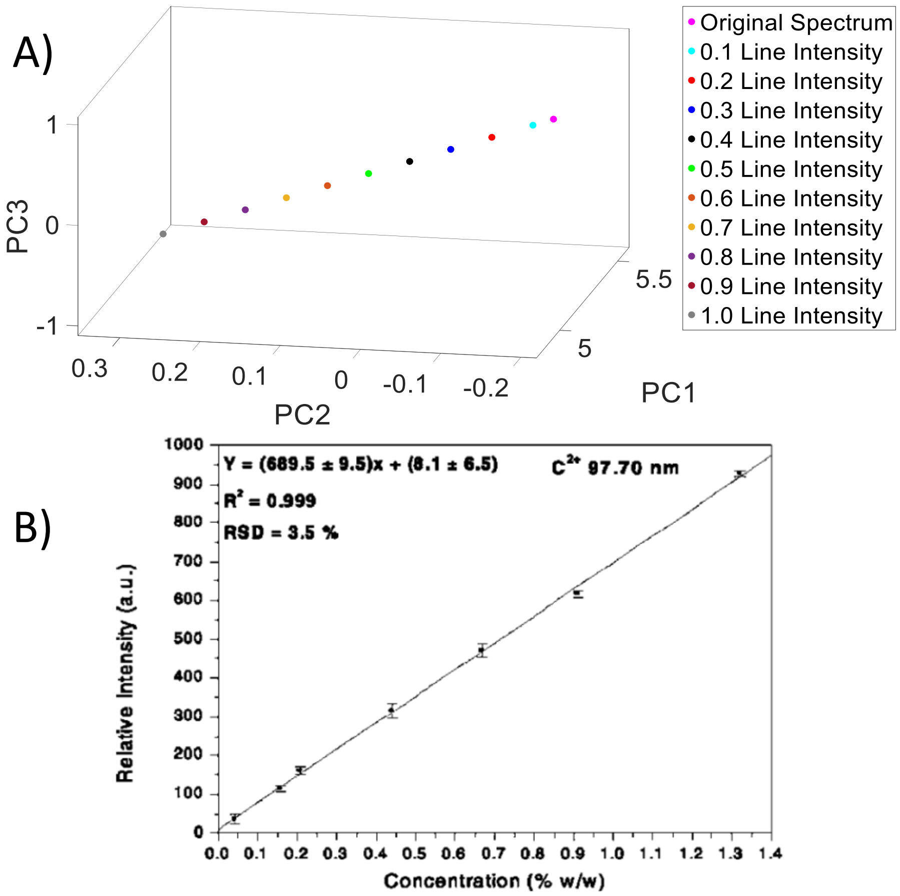


Figure 4.9: A) 3-D PCA showing linear deflection as a function of concentration and B) typical calibration curve [15].

4.3 Initial machine learning analysis

Machine learning was implemented with the goal of classifying and distinguishing between 4 significantly differing tablets. The purpose of this investigation was to investigate how well each technique performed, both against each other and against itself in different wavelength regions. 12 normalised single shot spectra, considered individually not as an average, were used per sample per wavelength region (45 nm, 75 nm , 85 nm, 95 nm and 105 nm). PCA, as visualised in figure 4.7, informed the analysis by providing a visual preliminary measure of how separable spectra for each sample would be within each wavelength region. The following techniques, as outlined in chapter 3 and using the codes in appendix C, were used to analyse the experimental spectra for the 4 samples: SOM [16], SVM [17] and CNN [18].

Two different test arrangements were performed using the experimental data presented in section 4.2.1. The first, referred to as a “1 vs 1” test, where the learning algorithms, once trained, decided whether an unknown test spectrum came from sample A or B.

For this test every combination of two classes (i.e., sample 1 vs sample 2, sample 1 vs sample 3, sample 1 vs sample 4, sample 2 vs sample 3, sample 2 vs sample 4 and sample 3 vs sample 4), was evaluated, while the data presented here are a consolidation for the total results. A driving force behind the implementation of the 1 vs 1 test is that SVM, as used in this study, can only be performed in this type of test. A 1 vs 1 test is a binary test, meaning there are only two classes involved so a 2x1 SOM arrangement is adequate for evaluation.

The second type of testing was labelled a “multiclass” test, where the learning algorithms, once trained, decided whether a test spectrum came from sample 1, 2, 3 or 4.

For both tests the experimental spectra from a sample were randomly separated into 2 groups, 8 spectra for training and 4 spectra for testing. Once an algorithm was trained the 4 test spectra were processed and the number of correctly classified spectra recorded (i.e., the number of spectra which were correctly identified as the correct sample type).

To ensure that the results were not influenced by the assignment of individual spectra to a particular group the random assignment of training and testing data was repeated 5 times to give a better indication of the overall performance. This resulted in a total of 240 spectra for 1 vs 1 testing and 80 spectra for multiclass testing. The overall percentage of correct classification for the 1 vs 1 testing is shown in table 4.1 for each technique. The corresponding percentages for multiclass testing are shown in table 4.2. The data present in tables 4.1 and 4.2 are reported to two decimal places, however, for 1 vs 1 and multiclass testing 240 and 80 spectra were evaluated, respectively. It is therefore to be expected that only differences in performance above 0.4% and 1.25% could be considered significant in tables 4.1 and 4.2 respectively.

For the SOM analysis a number of different layouts of the neural network architecture were used. The nature of each sub-layout will be identified as SOM axb, where a and b are the number of neurons initialized in each direction (e.g., SOM 5x5 consists of 25 neurons organized in a 5 by 5 grid). The SOMs use the experimental spectra to both train the network, by grouping the spectra onto individual neurons, and to test its performance, by checking if spectra that were not used for training are subsequently assigned to neurons associated with the correct sample type.

For the SVM analysis two different kernel functions were used, a linear kernel and a radial basis function (RBF) kernel as described in section 3.4 [19]. Both methods require some pre-processing of the experimental data: the first two PC scores for all spectra to be analysed are used as the input. These pairs of scores for each spectrum are again divided into two batches: one for training and one for testing. For validation k-fold (where k = 4) cross validation was used to designate validation sets, while Bayesian optimization was used to find the optimal hyperparameters to enhance the training of the network. The test batch was used to test the SVM.

Finally, a 7-layer CNN was also investigated. The method also required pre-processing of the experimental data, as outlined in section 3.3. A short time Fourier transform was performed on each spectrum to produce an image, which was then dimensionally reduced to a 50x50 pixel image. These images, corresponding to each spectrum, were used as the input for the CNN. The input images were divided into training, validation and testing batches. The 8 spectra assigned to training were split into two sub batches: 6 spectra to train the network with and 2 to validate with (testing during training). The spectra in these sub batches were shuffled once. The CNN is trained and validated using the stochastic gradient decent with momentum method, again with the aid of Bayesian optimization. Once trained the testing batch is used to assess performance.

Table 4.1: Percentage accuracy of classification for SOMs, SVMs and CNNs for 1 vs 1 testing.

	Wavelength Region				
Technique	**105 nm**	**85 nm**	**75 nm**	**45 nm**	**Average**
SOM 5x5	27.1	27.1	30.0	24.2	**27.1**
SOM 4x4	52.5	50.0	54.6	43.8	**50.2**
SOM 3x3	80.0	77.1	83.8	78.8	**79.9**
SOM 2X2	96.3	98.3	98.8	98.3	**97.9**
SOM 2x1	100.0	100.0	86.3	100.0	**96.6**
SVM RBF	93.3	97.1	98.8	98.3	**96.9**
SVM linear	91.3	95.8	99.2	97.9	**96.0**
CNN	99.2	100.0	99.2	99.2	**99.4**

From table 4.1 several key findings can be determined for the 1 vs 1 testing. From the averages across all wavelength regions the classification accuracy of the CNN outperforms the other techniques, although the performances of the SOM 2x1, SOM 2x2 and both SVMs are also quite good. As the dimensionality of the SOMs increase above 2x2 the performance starts to degrade. The network insufficiently clusters spectra to an increasing degree as the SOMs initial update radius remains fixed while the neuron layout grows beyond the 2x2 optimum configuration of this particular application.

Moving to the individual wavelength regions, a similar trend is observed. The performances of the SOM 2x1, SOM 2x1, both SVMs and the CNN are all strong. The SOM 2x1 performs best in most regions but is relatively poor in the 75 nm region.

These SOM 2x1 results are quite similar to the PCA results. Where it works it has exceptional performance, but the performance drops dramatically when it is not exceptional. The SOM 2x2 is very consistent across all regions, but the most consistently good performance across all regions is achieved by the CNN, indicating that it is the preferred technique.

Table 4.2 reveals the performance of the multiclass testing. Again, only the SOMs and the CNN were included in this analysis as the SVMs used were not capable of multiclass testing. Once more the CNN had the best average classification accuracy, this time followed by the SOM 3x3. An explanation for the larger optimal SOM size is that there are more classes in a multiclass arrangement as compared to the 1 vs 1 testing in table 4.1. The performance of the SOMs again decreases if an improper size is selected. If the SOM is too big or too small, it loses inter class separation. If a SOM is too small it loses its capability for inter class separation. The neurons are polluted, as all neurons within the update radius are updated. This leads to a loss in the networks capability to discriminate between different classes as training progresses.

Table 4.2: Percentage accuracy of classification for SOMs and CNNs for multiclass testing.

	Wavelength Region				
Technique	**105 nm**	**85 nm**	**75 nm**	**45 nm**	**Average**
SOM 5x5	68.8	63.8	76.3	73.8	**70.6**
SOM 4x4	88.8	85.0	91.3	91.3	**89.1**
SOM 3x3	92.5	97.5	100.0	97.5	**96.8**
SOM 2X2	31.3	51.3	50.0	55.0	**45.9**
CNN	98.8	100.0	100.0	97.5	**99.1**

4.4 Summary

This chapter outlined my initial experimental results and discussed the performance of VUV-LIBS for pharmaceutical analysis. The indications are that the spectra recorded from VUV-LIBS could prove a useful resource for the classification of pharmaceuticals. Overall, the best analytical tool was found to be CNNs, although under limited conditions each of the other methods could match, or better, the CNNs.

However, several issues also became apparent, both with the experimental arrangement and with the analytical implementations.

Most samples at some stage exhibited significant intra sample variations. Three possible experimental causes for these variations were identified:

1) Where very curved samples are studied it was observed that the spectral variations were decreased if the spectra were only recorded near the flatter centre portion of the sample.

2) When probing different sites on a sample surface the sample needed to be moved to present the new site to the laser pulse – if this distance was not sufficient an overlap of craters was observed again leading to increased intra sample variations. To overcome this a greater separation between sampling points is required.
3) Some samples exhibited significant fragmentation and delamination under the laser conditions used. This macroscale fragment ejection limitation has a dramatic effect on any subsequent spectra recorded from the initial site or close by.

The net result of these experimental issues was twofold: some large intra sample variations were inherent to the system and the number of sampling sites per sample was limited.

Some potential issues also arose with the analytical techniques. The first of these is directly related to the experimental issues. To better train the machine learning methods sufficient training data sets are required. However, the experimental issues could be a limiting factor for some of the methods i.e. insufficient spectra are generated to fully assess a particular method.

The second issue relating to the analytical method, save for the CNN, was the variation in classification success across different spectral regions and between different tests. This would indicate that knowledge of the physical system would be advantageous.

My results indicate that the acquisition of spectra from a wavelength region with several well-defined lines and little intra class variance is preferred when compared to capturing a larger spectral range at the cost of time and resources. If there are initial indications, such as good performance with PCA within a single wavelength range, that a region may provide the necessary information for classification analytics to operate on (and separate spectra into distinct classes) it would be optimal to capture more spectra in that region, than to try a probe a wider spectral range. Furthermore, due to the cross-correlation and massively iterative nature of the algorithms used, as described in chapter 3, providing a larger spectral range such as the one shown in figure 4.1 has the potential to in fact hinder classification accuracy as some regions may have high similarity between samples which the algorithms would learn, rather than focus on learning the differences. This of course could be overcome by increasing the complexity of algorithm, such as going from the 7-layer CNN to ResNet50, but that introduces its own set of constraints (e.g., run time and memory usage).

References

[1] Khater MA. Spectroscopic investigations of laser-produced steel plasmas in the vacuum ultraviolet (Doctoral dissertation, Dublin City University); 2001.

[2] Lu H. UV and VUV photoionization of some metal atoms and ions (Doctoral dissertation, Dublin City University); 2021.

[3] Lu H, Varvarezos L, Alli MB, Nicolosi P, Costello JT, Hayden P. The 5d→ 6p EUV photoabsorption spectra of Pb II and Bi III: evidence of excited states. Journal of Physics B: Atomic, Molecular and Optical Physics. 2020 May 5;53(11):115001.

[4] Lu H, Varvarezos L, Alli MB, Nicolosi P, Costello JT, Hayden P. Reply to Comment on “The 5d→ 6p photoabsorption spectra of Pb II and Bi III: evidence of excited states” by AN Ryabtsev. Journal of Physics B: Atomic, Molecular and Optical Physics. 2021 Feb 5.

[5] Lu H, Varvarezos L, Hayden P, Kennedy ET, Mosnier JP, Costello JT. The 5d-6p VUV Photoabsorption Spectrum of Bi+. Atoms. 2020 Sep;8(3):55.

[6] Neettiyath A, Alli MB, Hayden P, Vasa NJ, Sarathi R, Costello JT. Vacuum ultraviolet laser induced breakdown spectroscopy for detecting sulphur in thermally aged transformer insulation material. Spectrochimica Acta Part B: Atomic Spectroscopy 2020; 163:105730.

[7] Zehra SS, Nicolosi P, Costello J, Hayden P. Aluminium thin films depth profiling using LIBS. InJournal of Physics: Conference Series 2019 Jul 1 (Vol. 1289, No. 1, p. 012043). IOP Publishing.

[8] Zehra SS, Costello JT, Nicolosi P, Hayden P. Time-integrated and time-resolved VUV LIBS: a comparative study. InQuantum Technologies 2018 2018 May 21 (Vol. 10674, p. 106741H). International Society for Optics and Photonics.

[9] Jiang X, Hayden P, Laasch R, Costello JT, Kennedy ET. Inter-pulse delay optimization in dual-pulse laser induced breakdown vacuum ultraviolet spectroscopy of a steel sample in ambient gases at low pressure. Spectrochimica Acta Part B: Atomic Spectroscopy. 2013 Aug 1;86:66-74.

[10] Myakalwar AK, Sreedhar S, Barman I, Dingari NC, Venugopal Rao S, Prem Kiran P, et al. Laser-induced breakdown spectroscopy-based investigation and classification of

pharmaceutical tablets using multivariate chemometric analysis. Talanta. 2011;87(1):53–9.

[11] Noll R. Laser Induced Breakdown Spectroscopy Fundamentals and Applications Springer. 2012;

[12] Cremers DA, Radziemski LJ. Handbook of laser-induced breakdown spectroscopy. John Wiley & Sons; 2013 May 13.

[13] O'Leary EP. VUV laser-induced plasma spectroscopy for low level sulphur detection in steel (Doctoral dissertation, Dublin City University); 2007.

[14] Jiang X. Dual-pulse laser induced breakdown spectroscopy in the vacuum ultraviolet with ambient gas: spectroscopic analysis and optimization of limit of detection of carbon and sulfur in steel (Doctoral dissertation, Dublin City University); 2012.

[15] Khater M A. et al, Time-integrated laser-induced plasma spectroscopy in the vacuum ultraviolet for the quantitative elemental characterization of steel alloys, J. Phys. D: Appl. Phys. 33 (2000) 2252–2262.

[16] Pagnotta S, Grifoni E, Legnaioli S, Lezzerini M, Lorenzetti G, Palleschi V. Comparison of brass alloys composition by laser-induced breakdown spectroscopy and self-organizing maps. Spectrochim Acta Part B At Spectrosc. 2015;103–104:70–5.

[17] Dingari NC, Barman I, Myakalwar AK, Tewari SP, Kumar Gundawar M. Incorporation of support vector machines in the LIBS toolbox for sensitive and robust classification amidst unexpected sample and system variability. Analytical chemistry. 2012 Mar 20;84(6):2686-94.

[18] Chengxu LU, Bo WA, Jiang X, Zhang J, Kang NI, Yanwei YU. Detection of K in soil using time-resolved laser-induced breakdown spectroscopy based on convolutional neural networks. Plasma Science and Technology. 2018 Dec 27;21(3):034014.

[19] Raschka S. How to Select Support Vector Machine Kernels. 2016. [Cited 21/06/2020]. Available from: https://www.kdnuggets.com/2016/06/select-support-

5 Refined classification of pharmaceuticals using VUV-LIBS and machine learning

This chapter contains subsequent and more comprehensive vacuum ultraviolet laser induced breakdown spectroscopy (VUV-LIBS) studies of pharmaceutical tablets. The design and implementation of these studies arose by learning from, and adapting to, the limitations encountered in the study presented in chapter 4. The first study is based on varying the experimental conditions for the spectra I record, as compared to chapter 4, and these will be outlined in section 5.1. The same analytical techniques were used in this study, with competitive learning (COMP) and principal component analysis (PCA) classifiers also added to the tools available. The second study, again building from chapter 4, uses the same data as recorded for chapter 4 with the introduction of a process of data augmentation (section 5.2.3) to provide enhanced machine learning.

The spectra used for the first study are shown in section 5.3.1 for a comparison of samples with different active ingredients and in section 5.3.3 for a comparison of differing brands with the same active ingredient. A comparison of the signal to noise ratio (SNR) between the spectra gathered in chapter 4 and chapter 5 is shown in section 5.3.2.

Multivariate analysis for spectra gathered for the comparison of significantly differing tablets is shown in section 5.4.1 and for the comparison of differing brands with the same active ingredient in section 5.4.2.

The results of these studies are shown in section 5.5. The results of the data augmentation are shown in section 5.5.1, while in sections 5.5.2 and 5.5.3 we examine our proof of principle that VUV-LIBS in conjunction with machine learning can tell pharmaceuticals apart via classification. Firstly we test if a combination of VUV-LIBS and machine learning can tell distinctly different tablets apart (section 5.5.2). Secondly we test if the combination of VUV-LIBS and machine learning can distinguish between very similar analytes, i.e. two brands with the same active ingredient. In this case we attempt to distinguish between two brands of paracetamol (section 5.5.3). A summary of the results and conclusions drawn from the testing in this chapter is given in section 5.6.

5.1 Modification of experimental conditions

Following from the initial studies a number of issues were identified, such as the inhomogeneous nature of samples, significant height variations across samples (particularly curve surfaces), varying sample sizes and delamination and fragmentation of loosely compacted samples under laser pulse irradiation. To address these issues significant re-alignment and optimization of the experimental systems was required in order to: find stable positions of the samples, relative to the optical axis of the spectrometer, from which to record the spectra; the focusing conditions of the laser pulse onto the sample surface to prevent excessive damage of the surface; and determining the optimal spatial separations between sampling sites to prevent overlap of craters created by the laser produced plasma formation.

The options deemed most appropriate to address these issues are detailed in section 2.2.2 and so will be only summarized here. For all results presented in this section the vacuum in the chamber and spectrometer was maintained to a pressure of

~2×10^{-6} mbar. The 6 ns pulse from a Continuum Surelite laser was focused onto the samples to a spot size diameter of 200 µm. The large spot size diameter was chosen to ensure a large sampling site was taken to account for the sample inhomogeneity. The main difference in the study presented in this chapter is a reduction of the laser pulse energy to 265 mJ for the plasma formation. This was chosen to reduce the ejection of material from around the plasma formation site. This lower pulse energy yielded spectra with a lower and comparatively worse SNR, as discussed in section 5.3.2. The target surface was positioned closer now, at 0.4 mm from the optical axis, in order to produce stable spectra, as compared to the previous position in chapter 4 of 2 mm from the optical axis. The procedural work conducted to obtain these positions is detailed in section 2.2.2 and further evidenced in appendix E. For the study on significantly differing tablets the following tablets were analysed, a multivitamin, Disprin (containing aspirin), Panadol (containing paracetamol) and a sweet, which had a similar visual and surface appearance to the other samples. Each of these four sample types (as also used in chapter 4) define individual classes, which a test spectrum may belong to. The ingredients list and appearance for each of the samples used are given in section 2.3. The spectra I obtained for this study are shown in section 5.3.1. For the study on differing brands with the same active ingredient, Panadol and Panadrex were analysed, giving two classes. The spectra I acquired for the corresponding study are shown in section 5.3.3.

In both studies two wavelength ranges were investigated: at 65 nm and 105 nm central wavelengths (each wavelength range covered approximately 17 nm). Based on the rationale laid out in section 4.1, the 65 nm central wavelength range was selected as it was anticipated to yield high classification accuracy while the 105 nm central wavelength range was selected as it was estimated to yield comparatively lower classification.

The spectra used to generate the input for the various machine learning techniques could, in principle, be single shot spectra. However, in this study some simple elaborations, based on the improved set up, were employed. Two single shot spectra were obtained at a single site on the tablet in quick succession, with each stored as a separated “frame” with the same exposure time as for the spectra recorded in chapter 4. This automated acquisition is referred to as a kinetic series. The spectrum used in the analysis for a site is the sum of those two single shot spectra, normalised to the maximum intensity of the sum. The advantage of this mode of operation is that the single shot spectra can be retrieved at any time if required, whereas if both were recorded in a single long exposure only the summed spectrum is available. For each wavelength range 24 of these (summed and normalised) spectra are recorded from different sites on the surface for each sample.

This procedure was put in place as the individual single shot spectra were inherently weak but, due to the reduced laser pulse energy, more than one shot could be obtained at a single site before appreciable damage was observed and the intensity of spectra from a single site began to decrease, as seen in figure 5.1. As the first two shots produced consistent spectra they were used, with the summed spectra distinguishing the emission from the noise with greater clarity, as shown in figure 5.2.

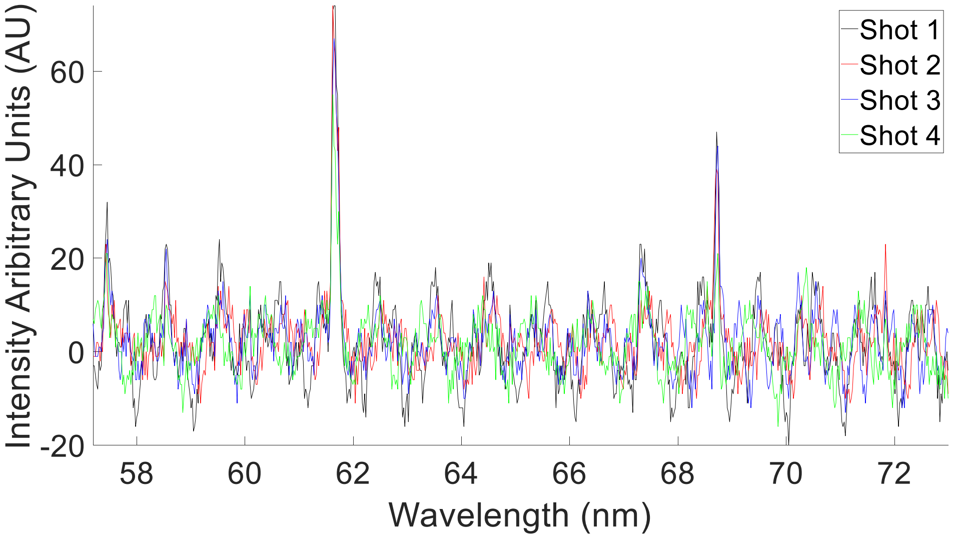


Figure 5.1: Four shots taken on the sweet target at the 65 nm central wavelength at a single site. Shots 1 and 2 are similar in intensity (especially for the strongest line), while the intensity starts to drop off for shots 3 and 4.

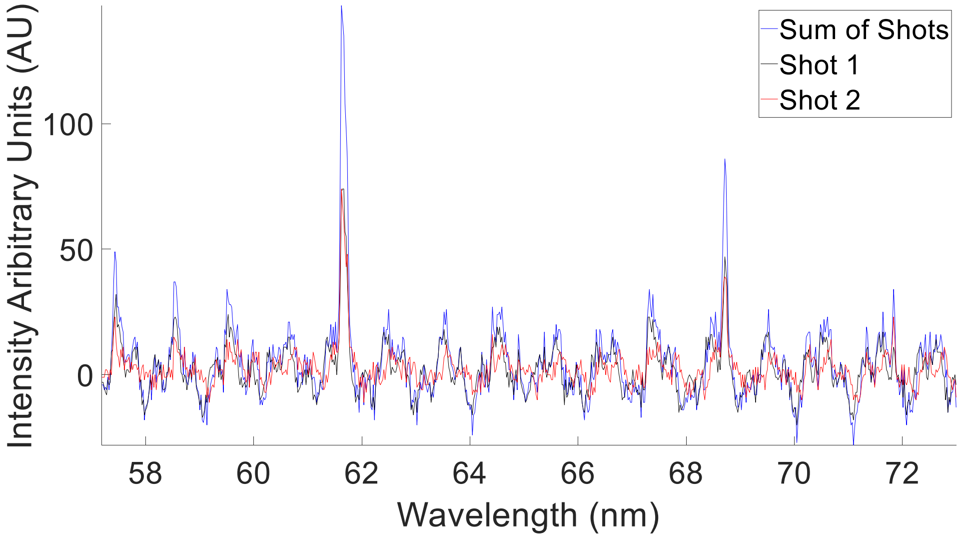


Figure 5.2: Spectra of shot 1 and shot 2, as well as the sum of those two shots, from a single sampling site. Notice how the intensity of the emission lines increase (doubles), but the noise in the spectrum remains almost constant.

5.2 Additional analytics

In addition to the experimental changes outlined in section 5.1, two additional analytical tools were utilised for classification. COMP, as described in section 3.3.1, was added to the suite of machine learning tools evaluated for classification fidelity. The network architecture used for COMP is laid out in section 5.2.1. PCA was re-configured to act as a classifier, taking it from a visualisation and pre-processing tool to a standalone classification technique. This allowed PCA to be evaluated quantitatively for its classification fidelity alongside the other techniques in this chapter. The methodology for this PCA implementation is described in section 5.2.2.

Aside from the addition of COMP and quantitative PCA as analytical tools, a study was undertaken to investigate the performance of data augmentation, using the data presented in chapter 4. With an understanding of the physical system being investigated, permutations of a small training data set were utilized to see if a larger augmented data set derived from that small experimental data set could yield an improved level of classification. Data augmentation was conducted as a separate study, augmented data was not utilized outside of sections 5.2.3 and 5.5.1 – all other sections of this chapter use exclusively experimental spectra.

5.2.1 Competitive learning

For the COMP analysis a number of different layouts of the neural network architecture were used, depending on the structure of the experimental study. COMP, as implemented here, operates on a layout specific to its task. For 1 vs 1 testing (where a decision is required to determine if a test spectrum belongs to class A or class B) two neurons are utilized. For multiclass testing, as there are four classes in question, four neurons are utilised. COMP uses normalized experimental spectra to both train the network, by grouping the spectra onto individual neurons, and to test its performance, by checking if a spectrum, is subsequently assigned to the neuron associated with the correct class. Additional details on the implementation of COMP are given in section 3.2.1 and section 5.5.2.

5.2.2 PCA classifier

Building on PCA as a utility, a classifier was also developed and utilized in sections 5.5.2 and 5.5.3. As with the basic visual analysis, spectra are defined to belong to a class if its associated data point, determined by its 3-D principal component (PC) score, falls within an ellipsoid, while not lying within other ellipsoids. Again, the ellipsoids are constructed such that the semi axes correspond to three times the standard deviation of the PC score in a given direction, for a given class/sample type. Computationally the decision boundary is drawn at the surface of the ellipsoid. Training data is used to populate and define the ellipsoid geometry. Test data which fall within an ellipsoid are classified as a particular class, while data points outside or on the surface of the ellipsoids are determined to be outside the class. The ellipsoids define a volume, to test if data points fall within that volume the inhull function developed by John D'Errico is used [1].

The benefit of introducing this automated PCA classification is a reduction in the time spent manually analysing figures. As studies scale, and perhaps many different analytes are compared, it becomes unfeasible to spend time analysing hundreds of graphs, and potentially many visual rotations of the 3-D plots, verifying their separability by eye and noting any overlaps or points outside ellipsoids. An unambiguous decision boundary and quick computation provide results faster. A percentage accuracy of classification can be calculated as the percentage of correctly classified spectra.

5.2.3 Data augmentation

Data augmentation, in machine learning, is the altering of base data [2-7]. Data augmentation is explored here as machine learning techniques typically benefit from training with large data sets to become generalized, as opposed to being over fit when training on a small data set [8-21]. A common example of data augmentation is used in image classification: in order to generate the invariance necessary for real world applications: rotation, zoom and cropping augmentations are commonly applied [22]. A key motivating factor behind using data augmentation for a given application may be that it is simply not possible to gather a large number of spectra/images typically used in machine learning for a given application (typically hundreds or thousands of images are used in the training process to account for all feature permutations).

A study on gait cycle analysis [23] noted that a small amount of data would be insufficient for training a convolutional neural network (CNN), which requires a large training data set to prevent overfitting. Data augmentation was therefore explored to generate a suitably large training data set.

A study on wearable sensor data for Parkinson’s disease monitoring using CNNs acknowledged that where possible, and available, CNNs will perform well when trained with large labelled data sets. However, in situations where only small-scale data sets are available it may be necessary to investigate data augmentation to accurately train the CNN. Application appropriate augmentations were applied, such as rotation (to simulate the rotation of the sensors in space), temporal permutations (to vary the location that an event takes place in within the time domain), jittering (to simulate additive sensor noise) and cropping (to diminish the dependency on event locations). These augmentations raised the baseline performance of the CNN classification from 77.54% to 86.88% when training with a larger augmented data set over the raw data set [24].

If the physical system is reasonably well understood, carefully made permutations of the base data can be applied and used to generate a larger training set [25]. A study on x-ray diffraction for thin film materials research outlined that the large amount of time spent in data acquisition is a limiting factor for research in that area. Three physics informed data augmentations were applied: peak scaling (to simulate the variation in peak intensity); peak elimination (to simulate the absence/presence of a line due to sample inhomogeneity); and pattern shifting (to account for the shifting of spectra due to hysteresis in the experimental setup). A CNN, with a mixture of augmented and experimental data, could achieve 92.9 % classification accuracy [26].

Given the success of data augmentation in other fields, an approach was designed to assess its applicability to the VUV-LIBS analysis of pharmaceutical tablets. Using the same spectra from the initial study (chapter 4) in the 105 nm region, the 12 spectra for each of the four tablet classes were divided into two groups. Two spectra were set aside per class to generate a larger training data set with data augmentation. The remaining ten spectra per class were reserved for testing. For the training data set three augmentations were applied: line removal; noise addition; and pixel shifts.

Line removal was carried out to simulate the presence/absence of an analyte due to inhomogeneity of the samples. An eleven-pixel segment, where the peak of the line of interest was at the centre of that segment, was removed. This was replaced with a segment consisting of eleven concatenated pixels, which consisted of the five pixels prior to the removed segment and the six pixels after the removed segment. With the line removal augmentation, the training set is doubled from two spectra to four per class (the original two, and two with a selected line removed). These four spectra per class as shown in figure 5.3 are then passed onto the next augmentation, which is noise augmentation. Note in figure 5.3 how similar the spectra for all the samples are, especially from the multivitamin and sweet (A and D, respectively). The original versions of both these differ by only a single line, while the relative intensity of the common lines quite similar. By generating the line removal augmentation, we incorporate how acceptable certain inhomogeneities are for the pharmaceutical classification. The advantage of CNNs is that we abstract and gather patterns to distinguish between pharmaceuticals, moving away from the analysis of isolated lines as is traditional in LIBS.

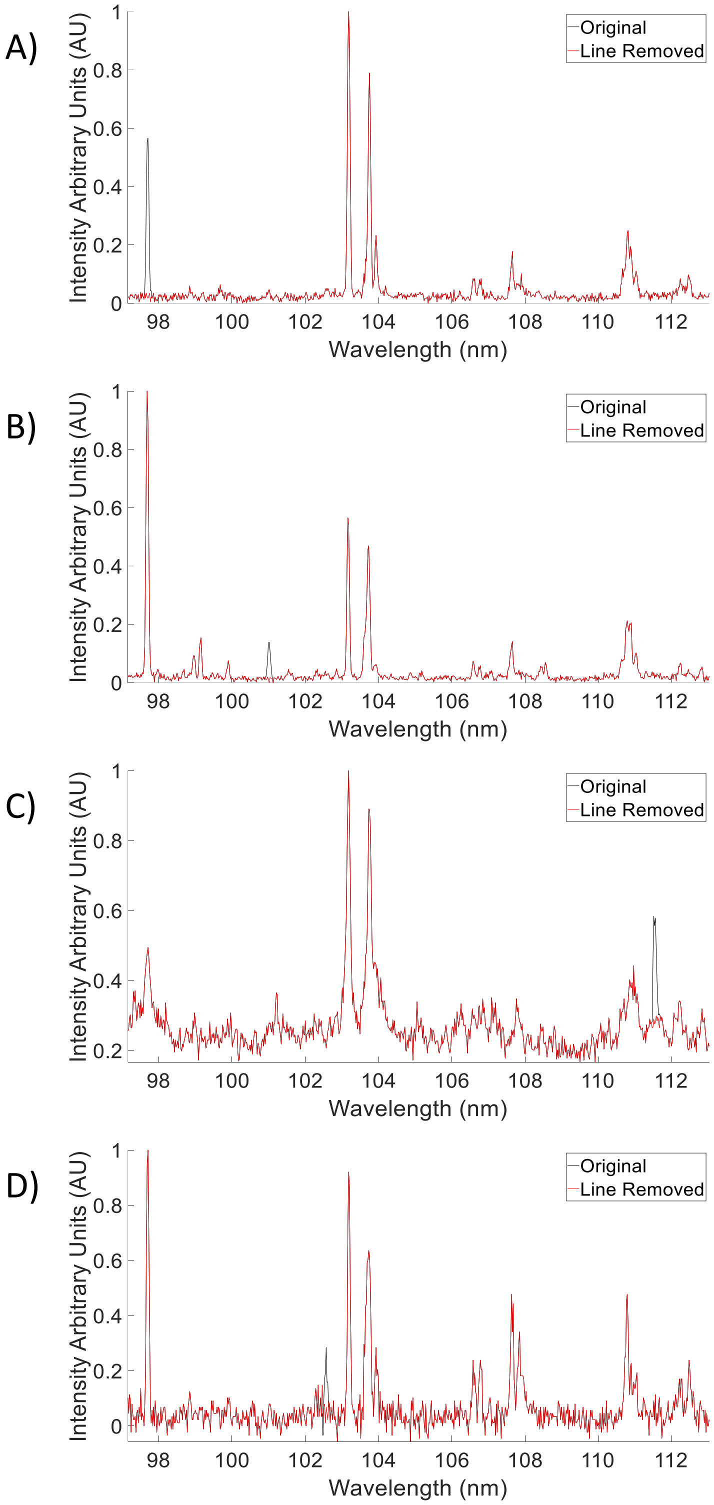


Figure 5.3: Examples of original (black) and line removed (red) spectra for various samples. A) multivitamin, B) Disprin, C) Panadol and D) sweet.

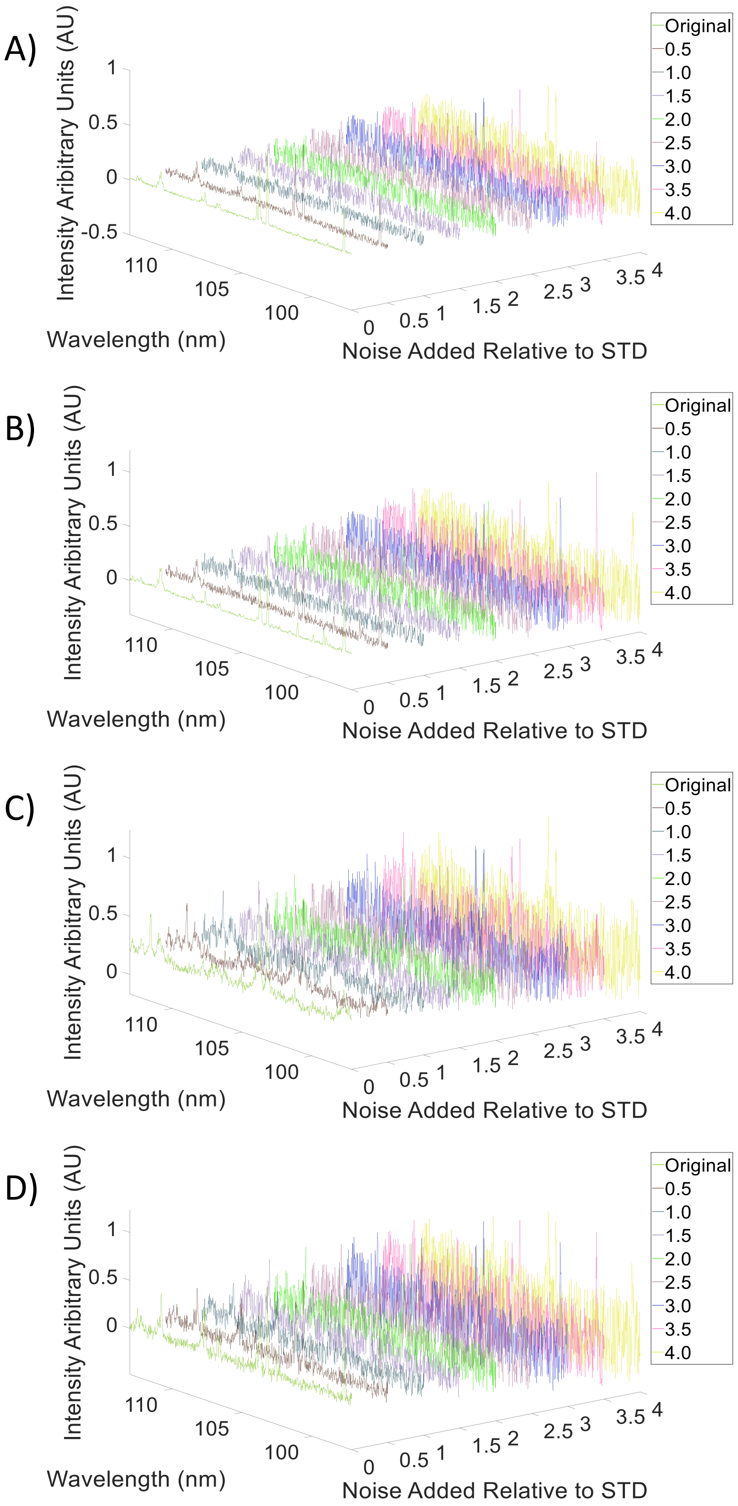


Figure 5.4: Example of original spectra and spectra with incremental noise additions (relative to the standard deviation) for A) multivitamin, B) Disprin, C) Panadol and D) sweet.

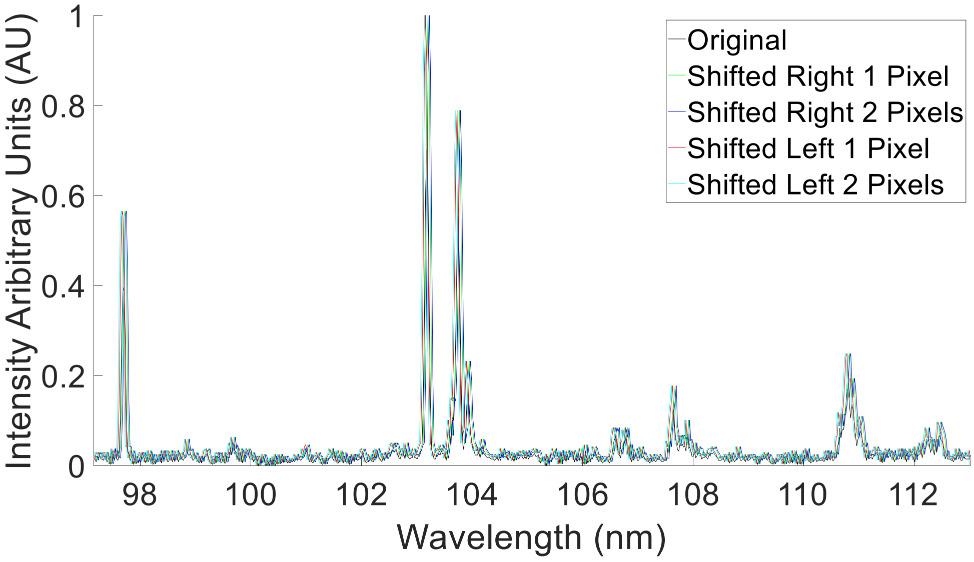


Figure 5.5: Visualisation of pixel shift data augmentation.

Noise addition is applied to simulate spectra acquired under a variety of noise profiles. Uniformly distributed white noise symmetric about zero, with its intensity dependent on a multiple of the standard deviation (STD) of the original input spectrum, was applied [27]. An example of this is shown in figure 5.4, with the original spectrum and incremental noise additions from 0.5 to 4 times the STD. Each of the four spectra per class from the previous augmentation now has nine different noise levels. 28 copies of these nine spectra are generated and these 252 spectra are passed to the next step in the augmentation process, which is pixel shifting.

Pixel shifting was introduced as a representation of the hysteresis of mechanically resetting the spectrometer to a previously selected wavelength value. Often a digital or analogue gauge will say it has been set to that previously explored value, but it can be a pixel (or more) off its stated setting. To represent this, each spectrum from the noise augmentation step, is randomly shifted left or right one or two pixels (or not shifted at all). A visual representation of this is shown in figure 5.5. Each of the 252 input spectra had four random shifts applied.

The net result of the data augmentation process was 1008 augmented spectra per class (with various combinations of line removal, noise level and pixel shifts) to train with (or 4032 spectra in total for all four classes). The results of using this augmented training data set are presented in section 5.5.1.

5.3 VUV spectra from pharmaceutical tablets

The VUV spectrum, from 37 to 112, nm recorded for each of the five tablets used in the study are presented in figure 5.6. To generate these broadband spectra 24 spectra recorded for each tablet in the 105 nm, 95 nm, 85 nm, 75 nm, 65 nm, 55 nm and 45 nm central wavelength regions were averaged. For convenience the different spectral regions will be referred to by their central wavelength for the remainder of this chapter (e.g., the 65 nm region refers to the region with a central wavelength of 65 nm), as a shorthand to ease the reading of this chapter.

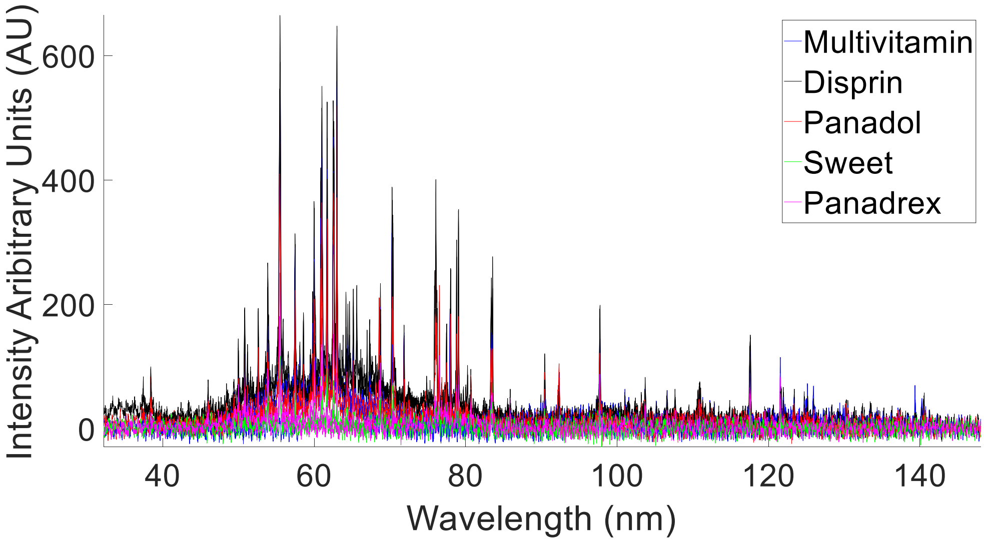


Figure 5.6: VUV spectra from the multivitamin, Disprin, Panadol, sweet and Panadrex tablets.

The spectra are visibly more overlapped and less distinguishable in figure 5.6, as compared to figure 4.1 in section 4.1. The significance of the spectra being more visibly overlapped implies that the spectra should be less separable for classification purposes, as compared to the spectra in figure 4.1. A greater degree of overlap between spectra from different tablets would indicate a lower degree of interclass variance, which should make distinguishing between analytes more difficult. The trade-off being made to generate the spectra here is that less energy is being imparted to the pharmaceutical tablet, making the investigation as minimally destructive as possible. This should allow for the gathering a larger number of spectra over the surface of a sample. While more spectra can be gathered, they appear to be less separable than when using a higher laser pulse energy.

5.3.1 Samples with different active ingredients

Looking at figure 5.7, which presents averaged spectra (of the 24 recorded) for the four tablets investigated, the 65 nm region represents a spectral range with comparatively more distinct spectral lines than the 105 nm region. In figures 5.8 – 5.11 the 24 difference spectra (the 24 spectra for each site subtracted from the mean for that class) for each tablet class in each region are shown. The 65 nm region in general possesses more spectral lines and a smaller spread in values of the difference spectra (indicating good overlap). Althought the largest flutucations in these difference spectra are concentrated about the common emission lines, these variations are inconsistent with no obvious trend within each sample type. There is a comparative lack of line emission in the 105 nm region and a larger intensity in the difference spectra (indicating poorer overlap) consitent across the entire spectral region, which would imply greater difficulty for the classification analytics to separate tables from one another using this data.

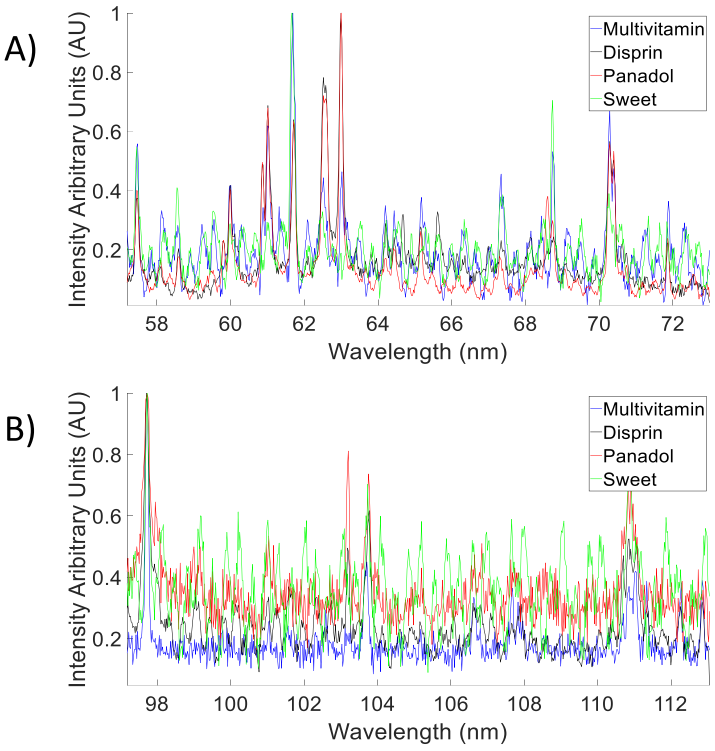


Figure 5.7: Averaged spectra for each tablet for various wavelength regions. A) 65 nm region B) 105 nm region.

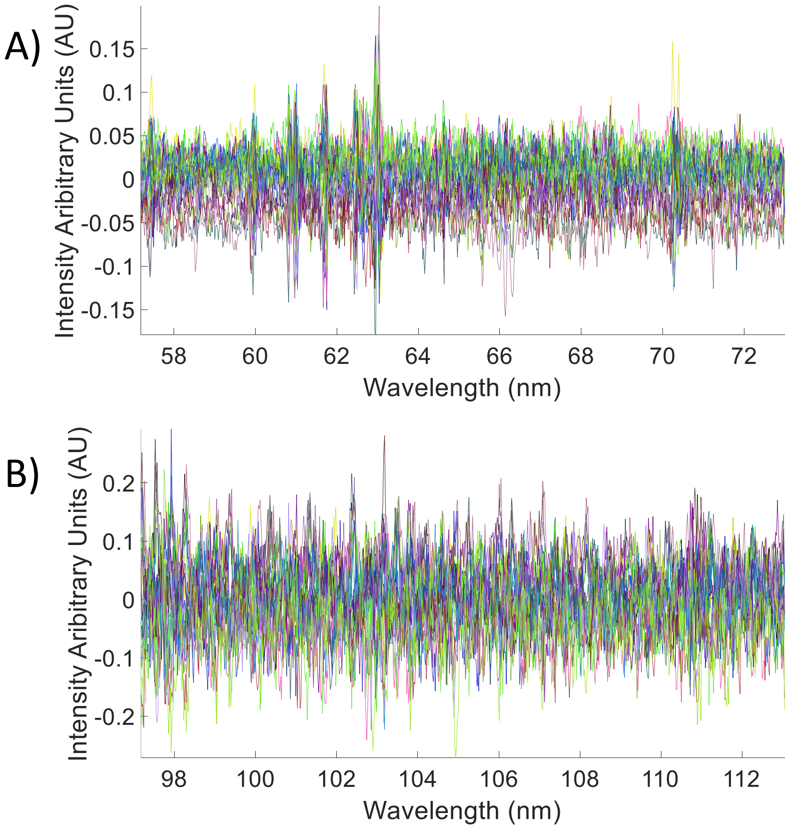


Figure 5.8: 24 difference spectra for the Disprin tablet in both regions. A) 65 nm region, B) 105 nm region.

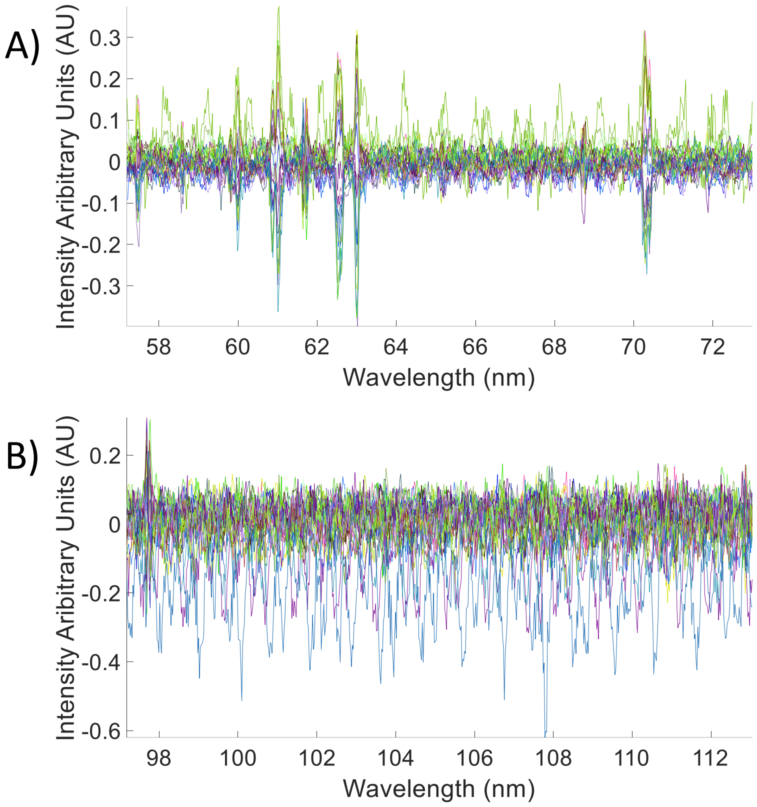


Figure 5.9: 24 difference spectra for the multivitamin both regions. A) 65 nm region, B) 105 nm region.

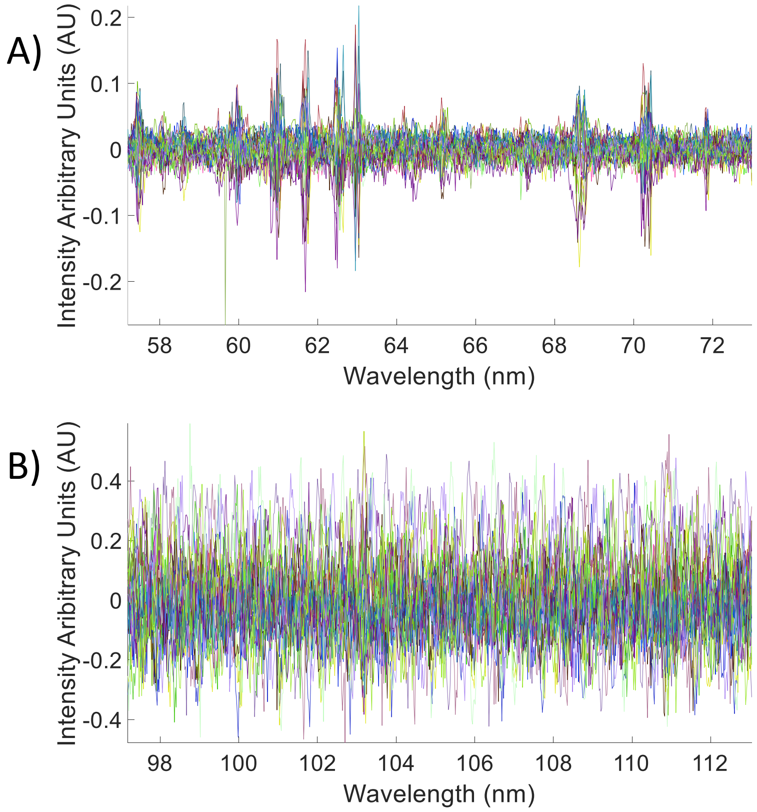


Figure 5.10: 24 difference spectra for the Panadol tablet in both regions. A) 65 nm region, B) 105 nm region.

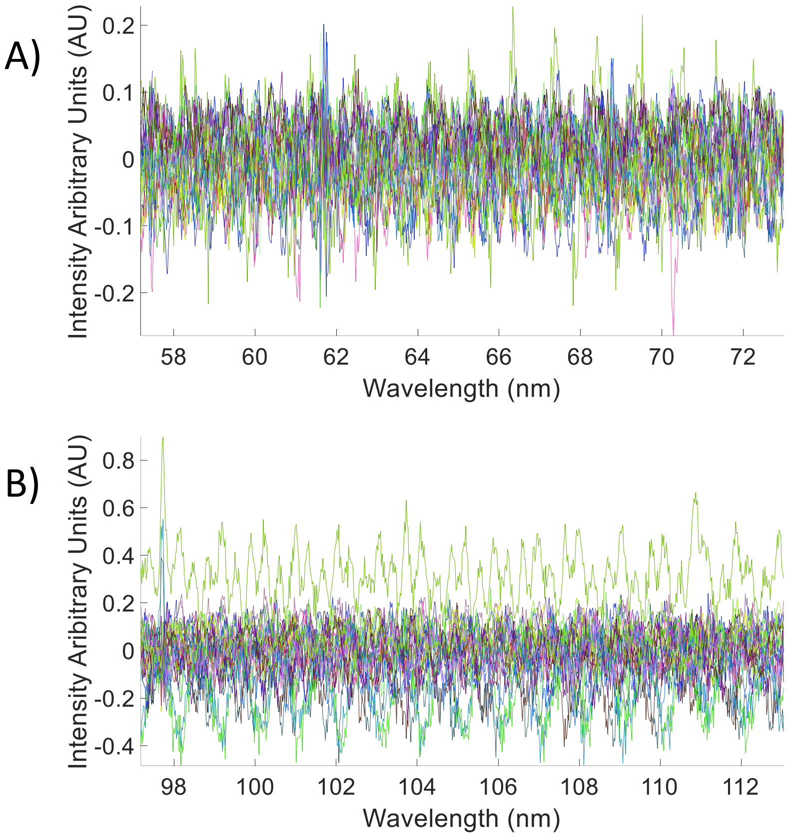


Figure 5.11: 24 difference spectra for the Sweet tablet in both regions. A) 65 nm region, B) 105 nm region.

5.3.2 Comparison of the signal to noise ratios

The average spectrum in the 105 nm region for each tablet from this chapter, and the initial study from chapter 4, are given in figure 5.12. To add some quantitative measures to the changes in the experimental data the SNR was calculated. The root mean square of 29 adjacent pixels (ranging from 97.92-98.56 nm) was used to determine the noise levels and divided into the peak value of the C III line at 97.703 nm to determine the SNR [28].

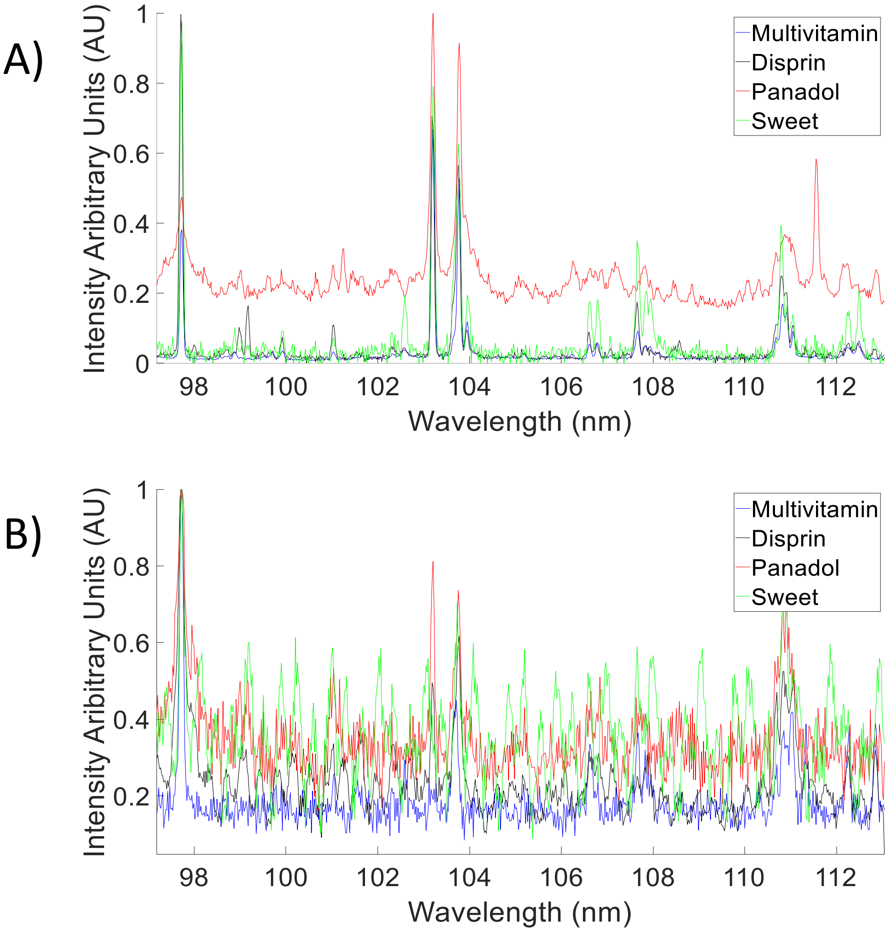


Figure 5.12: Wavelength region selected for SNR calculations. A) average spectra in the initial study. B) Average spectra in the current study.

Table 5.1: SNR of the average spectra for the comprehensive and initial studies.

Sample	Comprehensive study	Initial study
Multivitamin	6.4	28.2
Disprin	3.6	37.5
Panadol	2.5	2.1
Sweet	2.7	30.9
Study average	3.8	24.7

In both the comprehensive and the initial study, the Panadol tablet had the lowest SNR, as seen in table 5.1. It should be noted that this was the most curved of the samples studied. This curvature can lead to a number of issues, such as when firing on a curved tablet the laser optical path intersects diagonally with the sample surface, leading to the observed inconsistency of spectra recorded from the sample [29]; inconsistent plasma formation conditions due to varying lens–surface distances; inconsistent times/distances for the plasma to traverse before becoming visible to the spectrometer. However, it should be noted that in the comprehensive study the SNR actually improved slightly for this tablet, and the SNR achieved was similar to the SNR for the other tablets – with the revised experimental conditions it was on a par with the other (less curved) samples. This was achieved by only probing on the flattest section of the tablet and instead of 12 sites per tablet, 24 could be taken over the flatter part of the surface when factoring in intershot distances due to mechanical damage from shots and the size of adjacent melt zones.

When the study averages are compared, even with the Panadol bringing down the initial average considerably, the SNR for the initial study is still ~6.5 times higher than the comprehensive study. This is the trade-off with using the lower laser pulse energy for the comprehensive study: much more of the target can be probed (with spatial selectivity possible for "problem samples"), however the resultant spectra have a significantly lower SNR. In the previous chapter it was observed that good classification was possible for the initial study, however it remains to be seen if this is the case for the relatively poorer spectra obtained in the comprehensive study.

5.3.3 Differing brands with the same active ingredient

For this study spectra in the 65 nm region were recorded for two brands of paracetamol. The spectral region was chosen as this region appeared to have more defined spectral lines and a smaller intensity range of difference spectra, as discussed in section 5.3.1. The 24 difference spectra for each brand are shown in figure 5.13. Note the larger range of intensity values for Panadrex, especially around some of the spectral lines, indicating relatively greater overlap of the Panadol spectra and the comparatively poorer overlap of the Panadrex spectra. This suggests a large intra class variance in Panadrex, as compared to Panadol. Figure 5.14 shows the average of the 24 spectra for each sample. These averaged spectra, though different, do show a similarity in the abundance of lines at the same wavelengths, but at different intensities. Looking at section 2.3, we see these tablets differ by appearance and ingredients, so may also differ in terms of hardness and thermal conductivity.

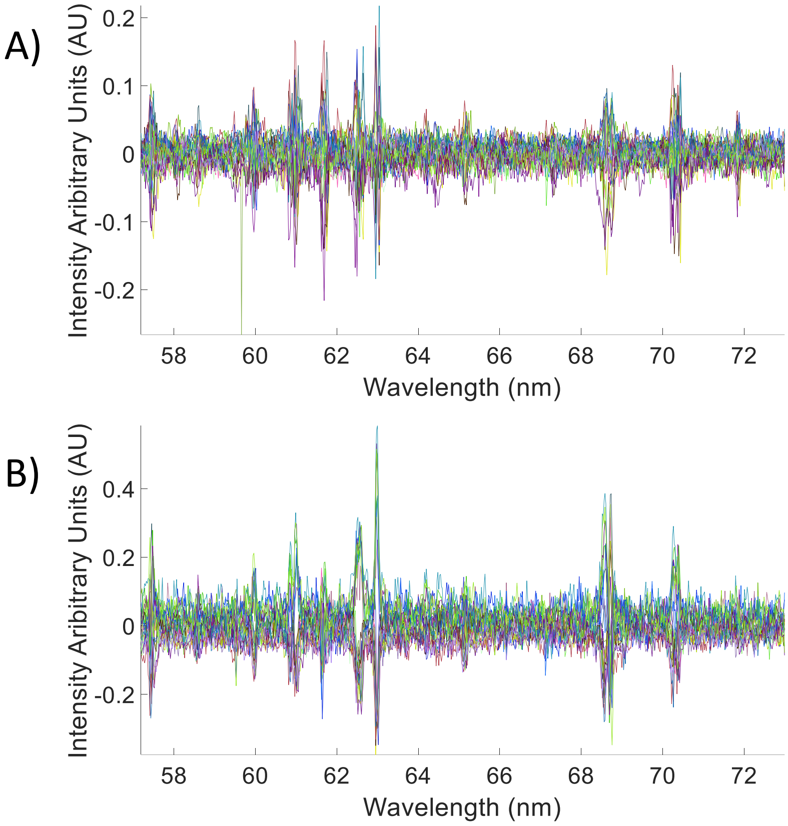


Figure 5.13: 24 difference spectra in the 65 nm region for A) Panadol and B) Panadrex.

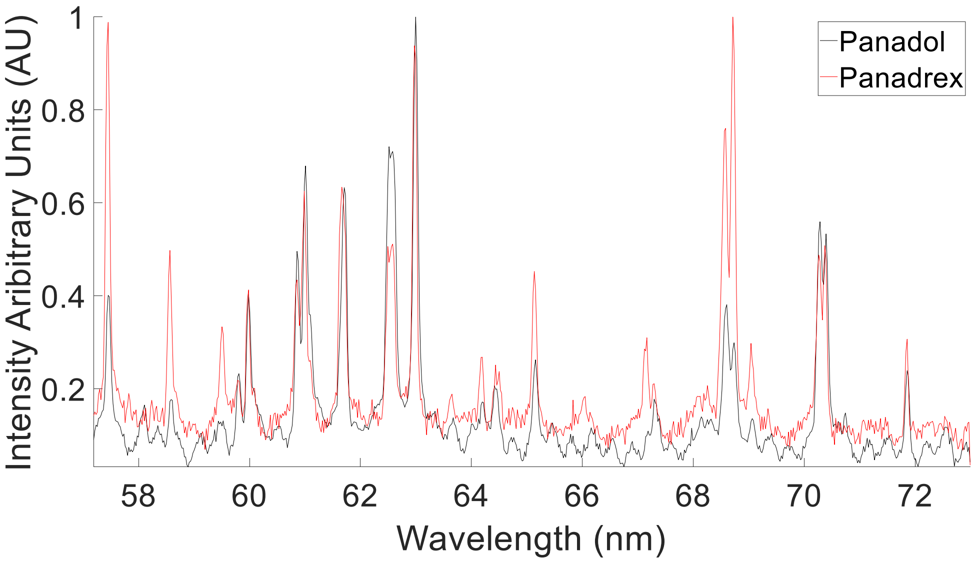


Figure 5.14: Averaged spectra in the 65 nm region for Panadol (black) and Panadrex (red) tablets.

5.4 Multivariate statistical analysis

As a first effort to classify the samples, the revised PCA technique outlined in section 5.2.2 (i.e., including the quantitative ellipsoid classifiers) was applied to the experimental data. This was done in turn for both the significantly differing tablets and the two brands with the same active ingredient.

5.4.1 Sample classification of significantly differing tablets

All 24 spectra per sample per central wavelength range were used for the PCA. The first three PC scores for each spectrum are used to represent the spectrum in a 3-D plot. The results of this analysis are shown in figure 5.15 A) and B). As previously encountered in section 4.2.1, the general trends of the averaged spectra and difference spectra (figures 5.7–5.11) are reproduced by the PCA. In figure 5.15 A) (showing the 65 nm region) there is less overlap of the ellipsoids representing each class, as compared to the 105 nm region shown in figure 5.15 B).

A greater intra class variance (as indicated by the larger ellipsoids) in the 105 nm central wavelength region is also observed in the PCA. This corresponds to the observations made when discussing figures 5.8–5.11. The intuitive reasons for this result remains the same: the greater the overlap of spectra recorded (or the narrow the spread in the difference spectra) within a sample the lower the intra class variance (indicated in PCA by small ellipsoids), whilst the greater overlap of spectra from different samples (similar looking spectra) the smaller the inter class variance (indicated in PCA by overlapping ellipsoids). This initial PCA would suggest spectra taken in the 65 nm region could be more separable with the classification analytics. The ellipsoids are not completely separated in figure 5.15 A) or B), implying there will be some difficulty in separation, but less so in the 65 nm region.

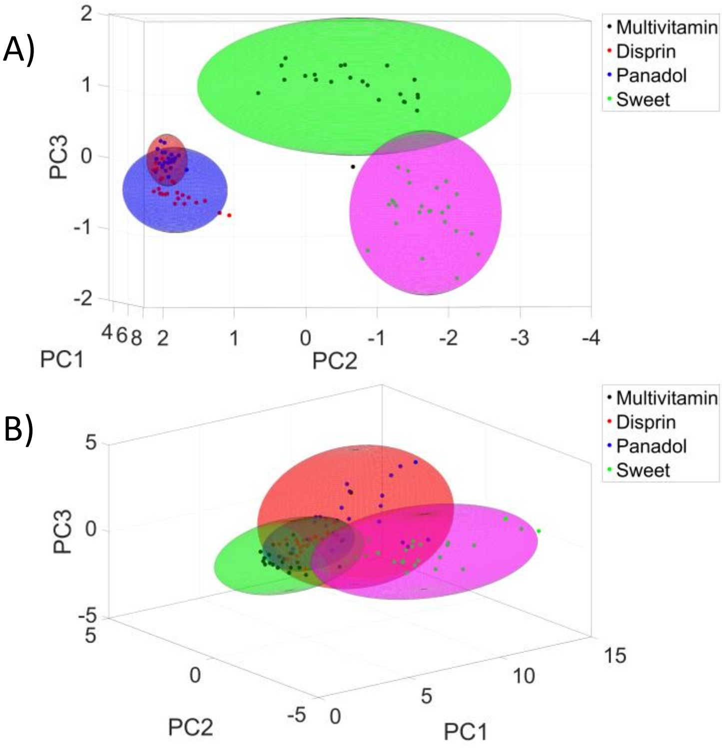


Figure 5.15: PC scores for each spectrum from each tablet type, and corresponding ellipsoid classifiers, at the A) 65 nm region and at the B) 105 nm region.

5.4.2 Sample classification of differing brands with the same active ingredient

In this study two brands of tablets with the same active ingredient of paracetamol (Panadol and Panadrex) were used. 24 spectra per brand were recorded in the 65 nm region (as this showed the most promise in section 5.4.1) for use in the PCA. Again, only the first 3 PC scores for each spectrum are used to represent a spectrum in the 3-D plot as shown in figure 5.16. In this figure only a small overlap of the ellipsoids representing Panadol and Panadrex is observed, while the Panadrex ellipsoid is much larger. This is again in keeping with the discussion of figure 5.13, where the larger range of intensity values for the strongest line implied a greater degree of variance with Panadrex spectra. Resulting in a wider distribution of datapoints in the PCA for the Panadrex.

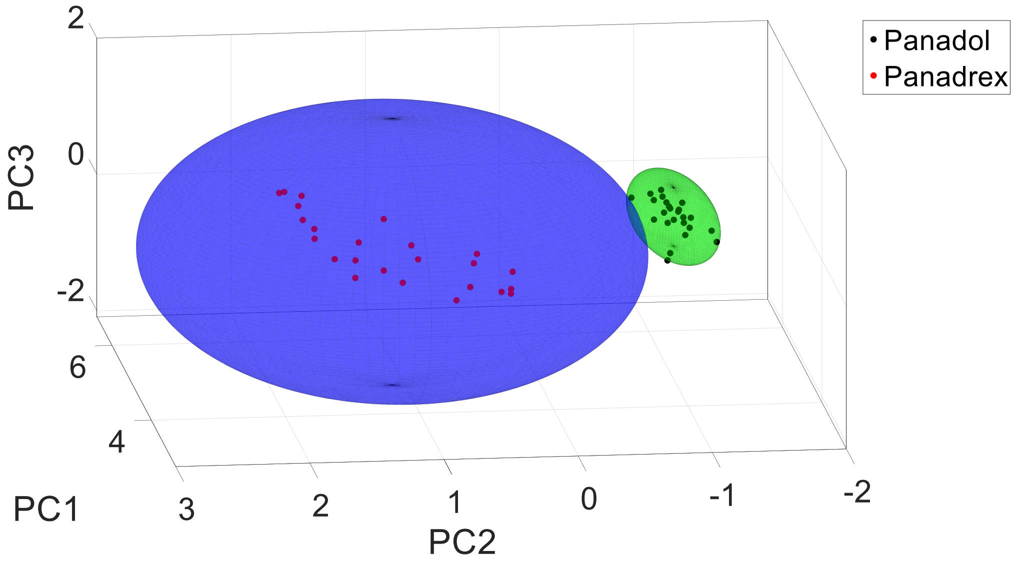


Figure 5.16: Panadol vs Panadrex: PC scores for each spectrum in the 65 nm region from each brand, and the corresponding ellipsoid classifiers.

5.5 Machine learning analysis

Machine learning, with the aim of correctly classifying test spectra, was performed on three different sets of experimental data. The first set of experimental data is from the initial study (see chapter 4) which consisted of four different types of tablets (the four "classes"). This data was used to assess the data augmentation process described in section 5.2.3, whereby a subset of the experimental spectra (two per class) was set aside and used to generate a large training data set via data augmentation (1008 per class, or 4032 in total). Once a CNN was trained with this data multiclass testing was conducted. All 40 of the spectra reserved for testing were evaluated to see which of the 4 classes they were assigned to. The results of this study are given in section 5.5.1.

The second set of experimental data used is discussed in section 5.3.1. In section 5.5.2 this data is used to evaluate the classification fidelity of the algorithms outlined in chapter 3 (with additional information in sections 4.3, 5.2.1 and 5.2.2) and using the codes in appendix C. The techniques used were COMP, self-organizing maps (SOM), support vector machines (SVM), CNNs and the PCA classifier, and the results for each are compared against one another.

The final set of experimental data to be classified, presented in 5.3.3, for two brands with the same active ingredient. These are also analysed using COMP, SOMs SVMs, CNNs and the PCA classifier, with the performance of each presented in section 5.5.3.

5.5.1 Sample classification with augmented data

In this study a 7-layer CNN (see section 3.2.4) was trained using spectra for the 105 nm region, which underwent the data augmentation process detailed in section 5.2.3. As the experimental spectra originated from four different types of tablet (a multivitamin, Disprin, Panadol and a sweet) there were four classes – one for each tablet type. After the data augmentation process there were 1008 augmented spectra per class and, as there were 4 classes, 4032 augmented spectra in total available for training purposes. During training those 4032 augmented spectra were split into a training and testing batch. Half the spectra were allocated to the training batch (to train the network) and half were allocated to the validation batch (to test the network during training). The spectra allocated to both batches were shuffled once. Ten experimental (unaugmented) spectra per class (40 in total) were available for multiclass testing, where the CNN, once trained, decided which one of the four classes a test spectrum belonged to.

The results of the testing are shown in table 5.2, with the percentage accuracy from the initial study in chapter 4 also included. 40 spectra were tested in the current study and 80 spectra were used in the initial study, only changes in performance above 2.5% and 1.25% respectively were considered significant as they constituted the percentage contributed by the classification of an individual spectrum in each case.

Table 5.2: Percentage accuracy of classification CNN results.

Data Augmentation	Initial Study
97.5	98.8

The CNN trained with 8 experimental spectra and data augmentation showed a high percentage accuracy of classification where only a single spectrum was incorrectly classified. The CNN trained with 160 real experimental spectra in chapter 4 also achieved classification where only a single spectrum was incorrectly classified. In effect there is no significant difference in the performance of the CNN method between the initial study and data augmentation, despite the 20-fold reduction in the amount of experimental data required to train the network. My results show that data augmentation can significantly reduce the number of experimental spectra needed to train a CNN.

5.5.2 Sample classification of samples with different active ingredients

As with the initial study, the machine learning techniques were employed to distinguish between the four classes of samples: a multivitamin, Disprin, Panadol and a sweet. The experimental data consisted of the 24 spectra recorded for each sample type, as presented in section 5.3.1. The spectra from each class were again randomly separated into a training and a testing group (16 and eight spectra, respectively, for each class). The random assignment to training or testing groups and the calculation of performance were carried out five independent times, in order to mitigate the influence of an individual spectrum belonging to either group. The results shown here are a consolidation of those five tests.

As in chapter 4, two testing arrangements were used. A 1 vs 1 arrangement where every combination of a binary test between two classes is conducted and consolidated, as well as a multiclass arrangement where a test spectrum was operated on to predict which of the four classes it belonged to.

For the COMP two neurons were defined for the 1 vs 1 testing, and four for the multiclass testing. For the SOMs the same layouts for the neural network architecture were used as chapter 4, i.e., SOMs of the type labelled SOM axb, where a and b denote the number of neurons in each direction. It should be noted that for both COMP and the SOMs the test can be failed in two ways: the test spectrum is assigned to a neuron associated with the incorrect class or the test spectrum is assigned to a neuron associated with more than one class.

For the analysis using SVMs both linear and radial basis function (RBF) kernels and two k-fold cross validation types (k=4 and k=32) were used to designate a validation set. Before starting the SVM analysis PCA was performed on the experimental spectra, with the first two PC scores providing the input to the SVMs.

A seven-layer CNN was again used, where the input to the CNN required the generation of images from the experimental spectra. This was achieved by performing a short time Fourier transform (STFT) on the normalized experimental spectra to produce images and then dimensionally reducing those spectra to 50x50 pixel images. For the

CNN the 16 spectra assigned to the training batch are split into two sub batches, a validation batch (with 4 spectra, used to test during training) and a training batch (12 spectra used to train the network). The spectra are shuffled once between training and validation batches for network training.

It is worthwhile to note again the reason that the 65 and 105 nm regions were chosen for further analysis. The 65 nm region was chosen as it is expected to lead to high classification, while the 105 nm is expected to lead to low classification, based on a visual inspection of the experimental spectra.

The final performance of each type of analysis is given in table 5.3 and table 5.4 for both the 1 vs 1 and multiclass tests, respectively. In these tables the column labelled "average" is the average performance across both wavelength ranges. In table 5.3 the k-fold average is the average of the linear and RBF kernel averages (again, as implemented, the SVM is not capable of multiclass classification). The data presented in tables 5.3 and 5.4 are reported to one decimal place. For 1 vs 1 and multiclass testing 480 and 160 spectra were evaluated, respectively. It is therefore to be expected that only differences in performance above 0.2% and 0.6% could be considered significant in tables 5.3 and 5.4, respectively. The average in the tables is a measure of performance of a machine learning technique over both a good and bad wavelength region. This is an indicator of a technique's ability to operate over a randomly assigned spectral region, rather than one where initial analysis (e.g. PCA) has taken place to select an optimal wavelength region for classification.

Table 5.3 shows little difference in the performance between the 65 nm and 105 nm regions for the SVM model with k = 4, however the 65 nm performs slightly better, with the RBF kernel slightly outperforming the linear kernel. When k is increased to 32 there is a greater difference in performance between the 65 nm and 105 nm regions and, in this case, the RBF kernel outperforms the linear one. However, for all cases tested, the lower k-fold validation method provides higher percentages of correct classification, in some case quite significantly. The k-fold of 32 acts as a "leave one out" cross validation that can train the networks to fit a small subset of spectra quite well but when the networks then operate on many spectra the performance is poor. The over-trained networks lack generalization, resulting in inferior classification as compared to the k-fold of 4 networks.

In table 5.3 the SOMs have, perhaps, the most nuanced behaviour of all the machine learning techniques. Even though 1 vs 1 testing is occurring, the best performance is achieved with the use of between 4 (SOM 2x2) and 9 (SOM 3x3) neurons and in all neuron arrangements used the 65 nm region performs better than the 105 nm region. This indicated that, in the SOM architecture, either the optimal amount of feature space for clustering spectra is greater than the number of classes (i.e., a single neuron is not sufficient to "collect" all training spectra from a single class), or, due to the inherent update radius used for SOMs, cross contamination of neurons can take place (in the 2x1 structure all neurons are updated for all spectra). When the dimensionality of the neuron layout is increased, above the optimum (e.g., 3x3 as shown in table 5.3), the performance starts to decline as the networks ability to cluster spectra over the increasingly larger neuron layout with an update radius of fixed initial value across all testing becomes insufficient.

The PCA has the worst maximum performance for the 65 nm region and works poorly for the 105 nm region. However, it does perform similar to some of the SVM and SOM architectures presented in table 5.3. This is surprising given that no learning takes place in this rather simple geometric approach.

Table 5.3: Percentage accuracy of classification for SVMs, SOMs, PCA, COMP and CNNs, for 1 vs 1 testing at the 65 nm and 105 nm regions.

	Wavelength range			
Technique	**65 nm**	**105 nm**	**Average**	**k-fold**
SVM linear	88.8	81.3	**85.0**	**4**
SVM RBF	89.6	90.6	**90.1**	**4**
		k-fold average =	**87.55**	
SVM linear	84.6	66.7	**75.6**	**32**
SVM RBF	78.1	72.9	**75.5**	**32**
		k-fold average =	**75.57**	
SOM 2x1	94.6	36.3	**65.4**	
SOM 2x2	95.4	74.2	**84.8**	
SOM 3x3	92.1	79.8	**85.9**	
SOM 4x4	80.2	70.6	**75.4**	
SOM 5x5	54.2	47.9	**51.0**	
PCA	82.1	62.1	**72.1**	
COMP	96.3	39.2	**67.7**	
CNN	98.8	92.9	**95.8**	

COMP performs the worst overall for the 105 nm region, but very well in the 65 nm region. That is, it performs well when classification should be possible, but not so when classification is not expected. A comparison with the SOM 2x1 (as both contain 2 neurons) results highlight a key difference between the two learning techniques. While the results are similar, a slight improvement is observed for COMP.

This can be ascribed to the how the weights associated with a neuron are updated following the competition between neurons: in COMP only the winner updates (in this case a single neuron), however in SOMs the winner and neighbours are updated (in this case both neurons). This can then lead to crosstalk between the neurons, reducing the knowledge of the difference between classes.

The CNN, which contains the most detailed pre-processing of the experimental data to generate input images and the most involved architecture (7 layers), has the best performance across both wavelength regions (above 90% correct classification in both).

The average performance calculated for table 5.3 provides a convenient way to estimate the best method to use for 1 vs 1 testing if no information is known about the experimental spectra. COMP should not be used, as its results are quite variable. A SOM with a medium number of neurons can give reasonable performance, but not as good as a low k-fold RBF SVM. The best results, however, are achieved for the more involved CNN architecture. The advantage of the PCA method is not in absolute performance, but in its ease of implementation – it can easily be used as a guide to determine the level of sophistication required to perform classification through machine learning.

The results for the multiclass testing are given in table 5.4. Overall, the performance of the competitive learning techniques (SOM and COMP) are poor (averaging between 25% and 63% correct classification). There is also quite a difference between the expected good performance spectral region (65 nm) and the expected poor performance region (105 nm). While the percentage of correct classification reaches ~80% when 9 or 25 neurons are used for the SOM at the 65 nm, a maximum of ~33% is achieved for the 105 nm region. In general, for the SOMs when fewer neurons are used a drop-in classification is observed.

Table 5.4: Percentage accuracy of classification for CNNs, SOMs, PCA and COMP for multiclass testing at the 65 nm and 105 nm regions.

	Wavelength range		
Technique	**65 nm**	**105 nm**	**Average**
SOM 2x2	52.5	0.0	**26.3**
SOM 3x3	81.3	10.6	**45.9**
SOM 4x4	73.1	32.5	**52.8**
SOM 5x5	81.3	13.8	**62.5**
PCA	66.3	41.9	**54.1**
COMP	69.4	0.6	**35.0**
CNN	95.0	91.9	**93.4**

The COMP method used performed best for the 65 nm region, approaching a correct classification rate of 70%. It is worthwhile to note the difference in performance between the 2x2 SOM and the COMP results. Both methods utilize four neurons, and in an ideal case would both achieve 100% success, with one class per neuron. This is evidently not the case, but one might expect the performance across both to be at least equal – which is again not the case. The COMP method outperforms the SOM in both wavelength regions (~70% versus ~50% in the 65 nm region and ~1% versus ~0% again in the 105 nm region). The difference between the two four-neuron models can, in part, be ascribed the update radius used in SOMs, combined with the condition imposed that, for a correct classification, a test spectrum must be clustered to neurons associated with an individual class. Thus, classification of a test spectrum can fail if it is clustered to multiple neurons, some belonging to different class OR it is clustered to a single neuron, which is associated with multiple classes. This is most likely the case in the four neuron models at 105 nm, with poorer performance for the SOMs due to the initial update radius of three – almost ensuring cross contamination between neurons.

Once again PCA, while not a machine learning technique, unexpectedly performed well. Its average performance was better than COMP and most of the SOMs in this study and, although its peak performance was not as high in the 65 nm region, it outperformed both in the 105 nm region. PCA proved to be, yet again, a useful tool for visualisation and in some instances, classification.

CNNs, again, demonstrated the highest average performance of all techniques, and the highest peak performance in both spectral regions – achieving >90% correct classification. My results indicate that VUV-LIBS in conjunction with machine learning is capable of discerning between drugs containing different APIs using the whole spectrum approach.

5.5.3 Sample classification of differing brands with the same active ingredient

In the previous section the goal of the study was to classify samples with different active ingredients. In this section the goal is to distinguish between two brands, namely Panadol and Panadrex, with the same active ingredient, paracetamol. As only one drug was being tested against another, 1 vs 1 testing was conducted and evaluated in the better performing spectral region (i.e., the 65 nm region) from section 5.5.2. The SVM (k=4), SOM, PCA, COMP and CNN models previously described are used once again.

24 spectra in the 65 nm central wavelength region were taken per target, as described previously in section 5.3.3. For the following tests 16 spectra per sample were used for training and eight were used for testing. This random assignment of training and

testing data was repeated 5 times to give a better indication of the overall performance. A total of 80 spectra were evaluated, with the results presented in table 5.5 (it is expected that only a difference in performance above 1.25% is considered significant).

From table 5.5 it is seen that for the SVM both kernels performed very well – with only one incorrectly classified spectrum between them. The SOM configurations of 2x1 and 2x2 were capable of 100% classification. However, when the number of neurons is increased the classification accuracy drops off – reducing to ~50%.

Table 5.5: Percentage accuracy of classification for SOMs, SVMs, CNNs, PCA and COMP for 1 vs 1 testing of Panadol and Panadrex.

Technique	% Accuracy
SVM RBF k=4	**100.0**
SVM linear k=4	98.8
SOM 2x1	**100.0**
SOM 2x2	**100.0**
SOM 3x3	96.3
SOM 4x4	82.5
SOM 5x5	51.3
PCA	95.0
COMP	**100.0**
CNN	**100.0**

Once again PCA held its own against the more detailed methods, with a 95% correct classification rate. The COMP model shows excellent performance, achieving 100% classification when classifying the two very similar analytes. CNNs once again performs as well as it did in prior studies, also achieving 100% classification.

In this study it is observed that there is very little difference in performance when telling very two similar analytes apart between the machine learning techniques, provided they are properly configured. This study also shows that while the PCA approach to classification can be quite accurate, it has limitations for a given data set (due to merely populating a feature space based on similarity) and the iterative clustering capabilities of properly configured machine learning algorithms will outperform PCA. My results show the capability of machine learning to abstract subtle differences in the spectra and distinguish between two tablets with the same API with a whole spectrum approach. This study did not need to focus on a line / line of interest in the API, e.g., a metal like titanium or magnesium, to tell the two drugs apart. This demonstrates the viability of VUV-LIBS in conjunction with machine learning for the classification of very similar pharmaceutical samples, indicating the potential of this approach as a solution to PAT for at-line quality control.

5.6 Summary and conclusions

The data I recorded, and the machine learning approaches I implemented in this chapter, were designed as test, or proof of principle, to determine if VUV-LIBS in conjunction with machine learning can tell pharmaceuticals apart via classification. This was examined under two types of scenarios: distinctly different tablets and two brands with the same active ingredient. In both cases the results were very promising of the viability of this approach, with CNNs performing the best with 92%-100% classification accuracy obtained across all tests in different wavelength regions.

Additional key findings from the work presented in this chapter can be summarised as following:

1. **A lower pulse energy does not necessarily lead to a reduction of classification rates.**

   This study has shown that it is possible to use a lower laser pulse energy to minimise macroscale fragment ejection thereby probing more of the surface of the analyte tablets. While the leveraged spectra had a lower SNR they were still separable for the classification analytics in this thesis.

2. **Prudent data augmentation can significantly reduce experimental workloads, without reducing classification rates.**

   The data augmentation study has shown it is possible, when using good SNR spectra such as in chapter 4, that a small subset of spectra can be reserved for training. That subset can be used to generate a large training data set for machine learning, populated with variations based on an understanding of the physical system and yield a similar classification accuracy.

3. **PCA, with a quantitative classifier, can achieve similar success to more in-depth machine learning techniques.**

   PCA can be utilised both as an automated classification tool and a visualisation tool, though it was demonstrated that in most cases an optimally configured machine learning classifier outperforms it. However, when they were not optimally configured, PCA could outperform some of the machine learning classifiers. Also, PCA acts as an important indicator for the optimum spectral region for high classification testing – where PCA performs best the highest classification accuracy can be obtained with machine learning.

4. **CNNs performed the best (with >90% correct classification) across all tests in this study.**

   As seen from tables 5.3 and 5.4, CNNs (which were also the most complex technique employed) achieved the highest average percentage accuracy for classification for both the 1 vs 1 and multiclass testing on significantly differing tablets. However, in some circumstances it was possible for other machine learning techniques such as a SOMs, SVMs or a COMP (if those techniques are properly configured) to also achieve the same percentage accuracy for classification as a CNN (e.g., in the 1 vs 1 test shown in table 5.5).

My results presented in this chapter also indicate that SOMs must be configured with the correct number of neurons for a given application. Too many and the ability to cluster the training spectra is diminished, while too few and there is contamination of the neurons from multiple classes during training. COMP demonstrated better classification in the 1 vs 1 testing arrangement than for the multiclass arrangement. In multiclass testing COMP outperformed SOMs when significantly differing tablets were studied and the same number of neurons were utilised as shown in table 5.4. For SVMs the RBF kernel demonstrated higher classification of the spectra in this study than the linear kernel.

A study used LIBS to identify counterfeit pharmaceutical packaging [33], and also evaluated several machine learning techniques, namely random forests, naïve Bayes, a simple neural network, k-nearest neighbours, quadratic discriminant analysis and linear discriminant analysis. For the counterfeit study the data was dimensionally reduced, first by integrating lines of interest, then by using PCA with three PCs, and then those three PCs were used as inputs for classification. In binary (1 vs 1) testing these applications were incapable of 100% classification. The neural network performed the best of all the machine learning techniques. As shown in table 5.5 of this thesis, several machine learning techniques were capable of 100% classification, especially the CNN. In the counterfeit study [33] neural networks were once again the best performing techniques for multiclass results, but its implementation had a 13% average misclassification. In this chapter, looking at table 5.4, the CNN implemented in this thesis averages 93.4% correct classification. It should be noted that these two studies were carried out on different data, which may alter the success rates. However, additional explanations for the higher classification in this thesis are: the use of more sophisticated machine learning techniques, as described in chapter 3; the spectra in this thesis were normalized thereby reducing intra class variance and increasing inter class variance; the CNNs, as implemented in this thesis, operate on reconstructed whole spectra (by using the STFT, it was not required to reduce the dimensions of the input spectra), which may explain why classification was greater than in the counterfeit study, which operated on three PCs, thereby losing variance information as a result of dimensional reduction.

Another study analysing gunshot residue [34] using LIBS and machine learning held notable comparisons to the work reported in this chapter. Neural networks, logistic regression, naïve Bayes and a critical threshold method were compared. Binary testing was carried out with two experimental acquisition methodologies and neural networks performed the best of the machine learning techniques, achieving 100% classification in both of the two tests conducted. The critical threshold method only achieved 100% classification for one of the two tests and the remaining techniques were unable to achieve 100% classification at all. The critical threshold technique defined a spectrum to be within a class if its SNR was three times higher than the standard deviation of the background mean for each line of interest. This is a non-dimensionally reduced analogue for PCA classification as implemented in this chapter. The critical threshold method, however, does come with caveats. The feature engineering process of identifying lines of interest and setting data driven thresholds for a given application is a manual task, and in the gunshot residue study [34] it was shown that neural networks were the more robust analytical technique, achieving 100% classification fidelity across both tests.

A study on the classification of iron ore with LIBS and machine learning evaluated the application of k-nearest neighbours, neural networks and SVMs [35]. This study used spectra to generate a single wavelength scan spectrum, which was reduced to ten PCs and presented to the machine learning techniques for multiclass classification. K-nearest neighbours, neural networks and SVMs were capable of 82.96%, 93.33%, and 94.07% success rates, respectively. These results are impressive and exceed the average results for techniques outlined in table 5.4, but, when comparing to the results in the 65 nm region, the CNNs used in this chapter, when applied in a properly selected wavelength region, achieved 95.0% exceeding the iron ore studies successful classification rate.

While the work in this thesis is not the first in LIBS to use a CNN, it is believed to be the first to generate input images using a single acquired experimental spectrum. Traditionally CNNs (as they were developed for computer vision applications) require image-based inputs, overcoming the issue of shaping LIBS spectra into image inputs has seen creative solutions. One study used many time-resolved spectra loaded into a matrix and PCA to shape the spectra into an input image for CNN classification [36]. Another study took a wavelength scan consisting of many individual spectra and reshaped those spectra into an input image [37]. While these solutions solve the issue of generating an input image for CNN classification, they require the acquisition of many spectra to generate a single image. The work in this thesis requires only a single spectrum to generate an input image, utilising the STFT method outlined in section 3.3.

Neural networks can abstract and find patterns to discern between features in data provided to cluster them into independent classes [38-40]. The work in this thesis leverages the abstraction of neural networks. The application of the CNNs used for classification does not require specialist understanding to select lines, nor the selection of the appropriate number of PCs to use, rather the entire spectrum undergoes reconstruction as detailed in section 3.2.4 and is submitted for classification. This avoidance of data loss is a key trait which allows the high successful classification rates achieved in this work.

When considering the run times of all machine learning techniques, CNNs on the adhoc 7-layer network were the fastest machine learning technique taking less than two minutes and using by far the least memory, only 20 MB on the GPU. CNNs on ResNet50 typically took 8 days to run and using over 9.6 GB of GPU memory. A comparison of the classification capability of the ResNet50 network vs the adhoc 7-layer network can be seen in appendix B, where little difference is observed. SVMs took 12-20 minutes depending on the amount of data presented. SOMs took longer than SVMs once the number of neurons was greater than nine (3x3 neuron layout). The more neurons initialised the longer it took, typically over 30 minutes. COMP took longer than all size of the SOMs, this was due to the nature of the winner take all update mechanism.

All things considered, CNNs are by far the best choice of these techniques for the purposes here. This is due to their ease of use and quick run times, once an appropriate network was designed. On the negative side the network design was quite in-depth and non-trivial but was a once off investment of effort. In addition, larger scale training and classification, like the data augmentation study, could only be conducted in a reasonable timeframe with CNNs.

References

[1] D'Errico J. Inhull [Internet]. 2006 [Cited 03/04/2016]. Available from: https://www.mathworks.com/matlabcentral/fileexchange/10226-inhull

[2] Andrieu C, De Freitas N, Doucet A, Jordan MI. An introduction to MCMC for machine learning. Machine learning. 2003 Jan 1;50(1-2):5-43.

[3] Wong SC, Gatt A, Stamatescu V, McDonnell MD. Understanding data augmentation for classification: when to warp? In 2016 international conference on digital image computing: techniques and applications (DICTA) 2016 Nov 30 (pp. 1-6). IEEE.

[4] Aftab U, Siddiqui GF. Big data augmentation with data warehouse: A survey. In2018 IEEE International Conference on Big Data (Big Data) 2018 Dec 10 (pp. 2785-2794). IEEE.

[5] Hoyle B, Rau MM, Bonnett C, Seitz S, Weller J. Data augmentation for machine learning redshifts applied to Sloan digital sky survey galaxies. Monthly Notices of the Royal Astronomical Society. 2015 Apr 16;450(1):305-16.

[6] Martin M, Sciolla B, Sdika M, Quétin P, Delachartre P. Segmentation of neonates cerebral ventricles with 2D CNN in 3D US data: suitable training-set size and data augmentation strategies. In 2019 IEEE International Ultrasonics Symposium (IUS) 2019 Oct 6 (pp. 2122-2125). IEEE.

[7] Gan Z, Henao R, Carlson D, Carin L. Learning deep sigmoid belief networks with data augmentation. Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics. 2015 May 9 - 12;38:268-276.

[8] Li X, Yu L, Yang X, Ma Z, Xue JH, Cao J, Guo J. ReMarNet: Conjoint Relation and Margin Learning for Small-Sample Image Classification. IEEE Transactions on Circuits and Systems for Video Technology. 2020 Jun 29.

[9] Wang T, Su H, Li J. DWS-MKL: Depth-Width-Scaling Multiple Kernel Learning for Data Classification. Neurocomputing. 2020 Jun 16.

[10] Guo G, Dyer CR. Learning from examples in the small sample case: face expression recognition. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics). 2005 May 16;35(3):477-88.

[11] Shirahama K, Matsuoka Y, Uehara K. Video event retrieval from a small number of examples using rough set theory. InInternational Conference on Multimedia Modeling 2011 Jan 5 (pp. 96-106). Springer, Berlin, Heidelberg.

[12] Polson NG, Scott SL. Data augmentation for support vector machines. Bayesian Analysis. 2011;6(1):1-23.

[13] Lemley J, Bazrafkan S, Corcoran P. Smart augmentation learning an optimal data augmentation strategy. Ieee Access. 2017 Apr 24;5:5858-69.

[14] Ratner AJ, Ehrenberg H, Hussain Z, Dunnmon J, Ré C. Learning to compose domain-specific transformations for data augmentation. In Advances in neural information processing systems 2017 (pp. 3236-3246).

[15] Salamon J, Bello JP. Deep convolutional neural networks and data augmentation for environmental sound classification. IEEE Signal Processing Letters. 2017 Jan 23;24(3):279-83.

[16] Schlüter J, Grill T. Exploring data augmentation for improved singing voice detection with neural networks. In ISMIR 2015 Oct (pp. 121-126).

[17] Yu X, Wu X, Luo C, Ren P. Deep learning in remote sensing scene classification: a data augmentation enhanced convolutional neural network framework. GIScience & Remote Sensing. 2017 Sep 3;54(5):741-58.

[18] Le Guennec A, Malinowski S, Tavenard R. Data augmentation for time series classification using convolutional neural networks.

[19] Carneiro G, Nascimento J, Bradley AP. Unregistered multiview mammogram analysis with pre-trained deep learning models. In International Conference on Medical Image Computing and Computer-Assisted Intervention 2015 Oct 5 (pp. 652-660). Springer, Cham.

[20] Shirahama K, Matsuoka Y, Uehara K. Event retrieval in video archives using rough set theory and partially supervised learning. Multimedia Tools and Applications. 2012 Mar 1;57(1):145-73.

[21] Liu S, Deng W. Very deep convolutional neural network based image classification using small training sample size. In2015 3rd IAPR Asian conference on pattern recognition (ACPR) 2015 Nov 3 (pp. 730-734). IEEE.

[22] Mikołajczyk A, Grochowski M. Data augmentation for improving deep learning in image classification problem. In2018 international interdisciplinary PhD workshop (IIPhDW) 2018 May 9 (pp. 117-122). IEEE.

[23] Wu Z, Huang Y, Wang L, Wang X, Tan T. A comprehensive study on cross-view gait based human identification with deep cnns. IEEE transactions on pattern analysis and machine intelligence. 2016 Mar 23;39(2):209-26.

[24] Um TT, Pfister FM, Pichler D, Endo S, Lang M, Hirche S, Fietzek U, Kulić D. Data augmentation of wearable sensor data for Parkinson's disease monitoring using convolutional neural networks. In Proceedings of the 19th ACM International Conference on Multimodal Interaction 2017 Nov 3 (pp. 216-220).

[25] McFee B, Humphrey EJ, Bello JP. A software framework for musical data augmentation. In ISMIR 2015 Oct (pp. 248-254).

[26] Oviedo F, Ren Z, Sun S, Settens C, Liu Z, Hartono NT, Ramasamy S, DeCost BL, Tian SI, Romano G, Kusne AG. Fast and interpretable classification of small x-ray diffraction datasets using data augmentation and deep neural networks. npj Computational Materials. 2019 May 17;5(1):1-9.

[27] Mathworks. Rand [Internet]. 2019 [Cited 01/09/2019]. Available from: https://uk.mathworks.com/help/matlab/ref/rand.html

[28] Hahn DW, Omenetto N. Laser-induced breakdown spectroscopy (LIBS), part II: review of instrumental and methodological approaches to material analysis and applications to different fields. Applied spectroscopy. 2012 Apr 1;66(4):347-419.

[29] Dubey A, Keyvan G, Hsia R, Saranteas K, Brone D, Misra T, et al. Analysis of pharmaceutical tablet coating uniformity by laser-induced breakdown spectroscopy (LIBS). J Pharm Innov. 2011;6(2):77–87.

[30] Raschka S. How to select support vector machine kernels. 2016. [Cited 21/06/2020]. Available from: https://www.kdnuggets.com/2016/06/select-support-vector-machine-kernels.html

[31] Schwenker F, Kestler HA, Palm G. Unsupervised and supervised learning in radial-basis-function networks. In Self-Organizing neural networks 2002 (pp. 217-243). Physica, Heidelberg.

[32] Forestier G, Petitjean F, Dau HA, Webb GI, Keogh E. Generating synthetic time series to augment sparse datasets. In2017 IEEE international conference on data mining (ICDM) 2017 Nov 18 (pp. 865-870). IEEE.

[33] Haase EA. Validation of fast spectrochemical screening methods for the identification of counterfeit pharmaceutical packaging. (Masters Thesis, West Virginia University); 2020.

[34] Menking-Hoggatt K, Arroyo L, Curran J, Trejos T. Novel LIBS method for micro-spatial chemical analysis of inorganic gunshot residues. Journal of Chemometrics. 2019 Dec 9:e3208.

[35] Hao YY, Zhang L, Ren L. Application of Scikit and Keras libraries for the classification of iron ore data acquired by laser-induced breakdown spectroscopy (LIBS). Sensors. 2020 Jan;20(5):1393.

[36] Chengxu LU, Bo WA, Jiang X, Zhang J, Kang NI, Yanwei YU. Detection of K in soil using time-resolved laser-induced breakdown spectroscopy based on convolutional neural networks. Plasma Science and Technology. 2018 Dec 27;21(3):034014.

[37] Chen J, Pisonero J, Chen S, Wang X, Fan Q, Duan Y. Convolutional neural network as a novel classification approach for laser-induced breakdown spectroscopy applications in lithological recognition. Spectrochimica Acta Part B: Atomic Spectroscopy. 2020 Apr 1;166:105801.

[38] Koehrsen W. Feature engineering: what powers machine learning [Internet]. 2018. [Cited 06/08/2019]. Available from: https://towardsdatascience.com/feature-engineering-what-powers-machine-learning-93ab191bcc2d

[39] Filion A. Applied machine learning part 1: feature engineering [Internet]. 2018. [Cited 06/08/2019]. Available from: https://uk.mathworks.com/videos/applied-machine-learning-part-1-feature-engineering-1547849284703.html

[40] Filion A. Applied machine learning, part 3: hyperparameter optimization [Internet]. 2018. [Cited 06/08/2019]. Available from: https://uk.mathworks.com/videos/applied-machine-learning-part-3-hyperparameter-optimization-1547849445386.html

6 Conclusions and Future work

This chapter contains my conclusions of this thesis. It provides a general summary of the various experimental issues encountered in vacuum ultraviolet laser induced breakdown spectroscopy (VUV-LIBS) for pharmaceutical applications and how they were overcome. A summary of how the machine learning techniques performed compared against one another is also laid out. Future work regarding how more quantitative studies may be pursued and the use of dimensional reduction to facilitate higher dimensional neuron arrangements of competitive learning, to possibly achieve higher classification with that technique, are outlined.

6.1 Conclusions

This work is a proof of principle that VUV-LIBS in conjunction with machine learning can tell pharmaceuticals apart via classification. This work tested this principle in two ways. Firstly, by classifying pharmaceuticals that are very different from one another i.e., having different APIs. This classification was achieved, evidenced by the results in tables 5.3 and 5.4. Secondly, by classifying two different brands of the same pharmaceutical, i.e., paracetamol. The second test investigated the ability of machine learning to abstract and identify the differences between two pharmaceuticals with the same API and separate them. This second test opens the way for application of VUV-LIBS combined with machine learning as a solution for at-line analysis of similar analytes i.e., quality control. This classification was achieved, evidenced by the results in table 5.5.

The objectives of the research in this thesis were as follows:

- To optimize a VUV-LIBS experimental setup to capture spectra of various pharmaceuticals for analysis.
- To investigate the application of individual computational techniques capable of distinguishing between pharmaceuticals based on their spectra.
- To optimize and compare those computational techniques. By comparing the classification fidelity and the nuances of their implementation an optimal use case for each technique for pharmaceutical VUV-LIBS can be outlined.

The main innovations and contributions of my work are as follows:

- By lowering the pulse energy to gather more spectra over the surface of a sample. Although this yields spectra with poorer signal-to-noise, they can still be classified with machine learning analytics, as shown in the results of sections 5.1.1 and 5.1.2.
- This work is the first in LIBS to implement the use of a short-time Fourier transform (STFT) method to generate input images for a convolutional neural network (CNN) for VUV-LIBS spectra.
- The results of this thesis indicate that of all the machine learning techniques evaluated, CNNs have the best classification accuracy, along with that, the 7-layer network that I designed and used, had the fastest run time of all the machine learning techniques.
- This is also believed to be the first work in LIBS to carry out the development and application of an ellipsoidal classifier.
- Prudent data augmentation can significantly reduce experimental workloads, without reducing classification rates as shown in section 5.5.1.

When discussing data augmentation, the results in table 5.2 compare classification between a network trained with augmented data against a network trained with experimental data. It is observed that when experimentally gathered data was used to train a network 98.8% classification was achieved, when training with augmented data 97.5% classification was achieved. Both only had a single spectrum misclassified. There was effectively no difference in performance between the two, even though the data augmentation study used 20-fold fewer experimentally acquired spectra to train the network. This shows there is some merit in pursuing data augmentation in cases where there is a small amount of data available to populate a neural network for training, given there is a reasonable understanding of the physical system from which to generate those augmentations.

The capability of principal component analysis (PCA) to show a change in 3-D location when a single line is added to spectra, as well as how much the location changes with respect to the increase in the intensity of that line is demonstrated in figure 4.9. PCA performs well at tracking line intensities, while also, in conjunction with an ellipsoidal classifier, displayed promising sensitivity, and can, under the correct conditions, decide on whether spectra belong to a certain chemical sample set. However, in some circumstances more sophisticated analytics are required for classification, such as the machine learning algorithms implemented in this study. The usefulness of PCA is not limited to its functionality as another tool for classification. When used as a tool for visualization PCA can identify a spectral region with spectra that have a low intra class variance, as indicated by small ellipsoids. PCA can also identify spectral regions of greater inter class separability which is visualized by ellipsoids with ideally no overlap or less overlap than a spectral region with less separable spectra between classes.

Of all the machine learning algorithms evaluated in this thesis convolutional neural networks consistently performed the best, having the highest average classification both in multiclass testing and in 1 vs 1 testing, in both the initial study in chapter 4 and subsequent studies detailed in chapter 5, the results of which are shown in tables 4.1, 4.2, 5.3 and 5.4, respectively.

For support vector machines the radial basis function kernel yielded higher classification accuracy than the linear kernel when the k-fold was properly configured to k = 4. This behavior was observed for all studies where 1 vs 1 testing was conducted, as shown in chapters 4 and 5 in tables 4.1, 5.3 and 5.5.

Self-organizing maps (SOM) show a behavior that demonstrates it must be constructed with the appropriate number of neurons for grouping for a given application. SOMs are a technique that have historically been developed from pre-existing clustering methods such as k-means clustering [1-12]. K-means clustering takes place in a boundaryless feature space. The class each data point belongs to is dictated by the centroid it is closest to [13,14], as shown in figure 6.1. For successful classification via clustering the final placement of those centroids must be correct.

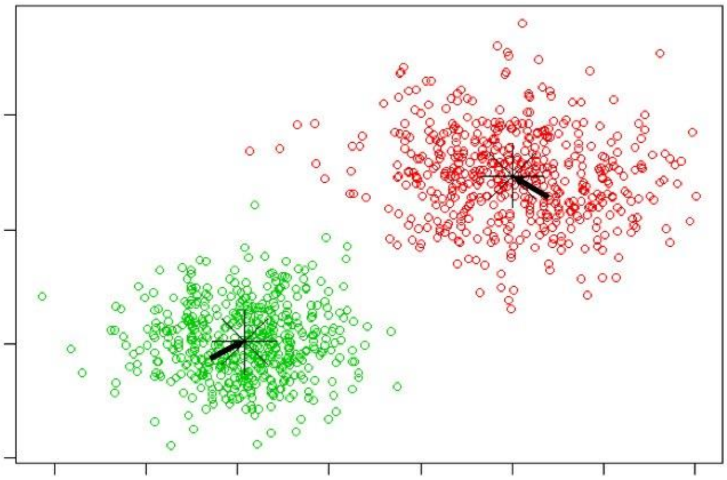


Figure 6.1: Visualization of k-means clustering, datapoints populate a boundaryless feature space [14].

Datapoints in SOMs can be considered bound in a competitive learning hypersphere, as shown in figure 6.2, where they cluster. Conceptually the volume of said hypersphere typically increases with the number of neurons [15]. Inter and intra class separation is affected by the number of neurons used.

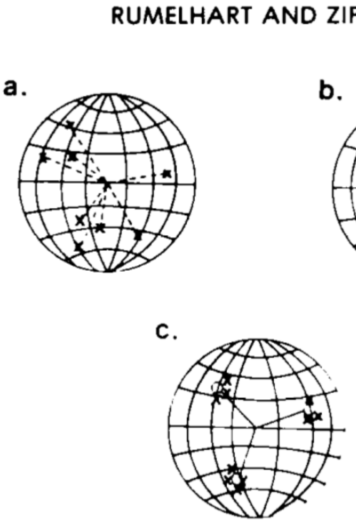


Figure 6.2: Visualization of the hypersphere feature space representation used for competitive learning applications. Spectra are collapsed to points and considered bound within a hypersphere. Points form clusters within a hypersphere as seen in sphere C [15].

For SOMs it stands to reason that with a decaying update radius and fixed feature space boundaries, that enough space (number of neurons) much be generated to allow separation between classes (inter class separation), but the boundaries must be small enough such that the update radius does not fail to cluster within a class (intra class separation). The number of neurons selected can be considered an optimization challenge, like the need for the correct placement of a centroid for k-means clustering.

6.2 Future work

The goal of the research was not a traditional quantitative or qualitative LIBS analysis, but rather the classification of pharmaceuticals into different classes. However, it is very clear that the physical, thermal and chemical properties of the pharmaceuticals are important as the initial conditions in a LIBS experiment will have a significant impact on the spectra recorded. As the samples were different commercially available products this introduced variability in the experiments that was unknown and uncontrolled. This was justified as the goal was to design and implement an experimental and analytical approach that could classify the differing tablets, rather than to perform a detailed study of what the spectral differences were or what gave rise to them.

Although this approach was successful, with the results in this thesis in table 5.5 indicated that detailed knowledge of the tablets nor API information were not necessary to achieve classification, a future study with greater control over the sample properties, such as generating test samples by controlled but varied compression of identical powders, would eliminate this variability and could provide insights into the origins of the features that are best matched to the VUV-LIBS with machine learning approach. This would also minimise variations observed in the spectra due to the differences in the samples structures and allow for chemical differences to be investigated for classification purposes.

A drawback of VUV-LIBS, within the context of this thesis, is the ejection of fragments on the macroscale, from the bulk pharmaceutical due to its grainy non-cohesive structure. Further investigating how to best ablate as little matter as possible per shot while still obtaining reliable spectra would demonstrate lager scale viability to the pharmaceutical industry for the application of VUV-LIBS.

Subsequent studies may benefit from quantitative investigations that focus on a single peak in a spectrum acting as an area of interest. Then that peak can be integrated, its integrated intensity can be likened to the available concentration of an analyte provided the corresponding concentration curves have been calculated.

Elemental maps [16-21] can then be plotted over a targets surface. If applied in conjunction with depth profiling that would allow for a visualization of the concentration and distribution (homogeneity) of the active pharmaceutical ingredient in a tablet at varying depths of its volume.

Building on the work in this thesis further pre-processing and arrangements of neurons can be investigated. Competitive learning (COMP) was given full spectra as their inputs, but dimensionally reducing those input spectra to 3 principal components would reduce the compute time and make higher neuron architectural arrangements processable within reasonable timeframes for a COMP architecture.

COMP was only tested in two configurations, using 2 and 4 neurons, where each class under examination is associated with a single neuron. Higher classification fidelity may be leveraged if COMP was evaluated with a higher number of neurons. To facilitate the transition from the current implementation to a higher neuron arrangement, the logic surrounding classification would adopt the same condition as SOMs. A class may associate with many neurons if the neurons are unique to each class.

References

[1] Hofmann T, Buhmann JM. Competitive learning algorithms for robust vector quantization. IEEE Transactions on Signal Processing. 1998 Jun;46(6):1665-75.

[2] Ahalt SC, Chen P, Chou CT, Jung TP. Implementation of a vector quantization codebook design technique based on a competitive learning artificial neural network. The Journal of Supercomputing. 1992 Feb 1;5(4):307-30.

[3] Card HC, Kamarsu S, McNeill DK. Competitive learning and vector quantization in digital VLSI systems. Neurocomputing. 1998 Jan 1;18(1-3):195-227.

[4] Yang TN, Wang SD. Competitive algorithms for the clustering of noisy data. Fuzzy Sets and Systems. 2004 Jan 16;141(2):281-99.

[5] Burke LI. Competitive learning based approaches to tool-wear identification. IEEE transactions on systems, man, and cybernetics. 1992 May;22(3):559-63.

Card HC, Rosendahl GK, McNeill DK, McLeod RD. Competitive learning algorithms and neurocomputer architecture. IEEE Transactions on Computers. 1998 Aug;47(8):847-58.

[6] Yair E, Zeger K, Gersho A. Competitive learning and soft competition for vector quantizer design. IEEE transactions on Signal Processing. 1992 Feb;40(2):294-309.

[7] Burke LI. Competitive learning based approaches to tool-wear identification. IEEE transactions on systems, man, and cybernetics. 1992 May;22(3):559-63.

[8] Anderberg MR. Cluster analysis for applications: probability and mathematical statistics: a series of monographs and textbooks. Academic press; 2014 May 10.

[9] MacQueen J. Some methods for classification and analysis of multivariate observations. In Proceedings of the fifth Berkeley symposium on mathematical statistics and probability 1967 Jun 21 (Vol. 1, No. 14, pp. 281-297).

[10] Ma J, Wang T. A cost-function approach to rival penalized competitive learning (RPCL). IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics). 2006 Aug;36(4):722-37.

[11] Banerjee A, Ghosh J. Frequency-sensitive competitive learning for scalable balanced clustering on high-dimensional hyperspheres. IEEE Transactions on Neural Networks. 2004 May;15(3):702-19.

[12] Kia SJ, Coghill GG. Unsupervised clustering and centroid estimation using dynamic competitive learning. Biological cybernetics. 1992 Sep 1;67(5):433-43.

[13] Joshi P. What Is K-Means Clustering? [Internet]. 2013 [cited 17/08/2019]. Available from: https://prateekvjoshi.com/2013/06/06/what-is-k-means-clustering/

[14] Jeffcock P. K-Means Clustering in Machine Learning, Simplified [Internet]. 2018 [cited 17/08/2019]. Available from: https://blogs.oracle.com/bigdata/k-means-clustering-machine-learning

[15] Rumelhart DE, Zipser D. Feature discovery by competitive learning. Cognitive science. 1985 Jan 1;9(1):75-112.

[16] Klus J, Pořízka P, Prochazka D, Mikysek P, Novotný J, Novotný K, Slobodník M, Kaiser J. Application of self-organizing maps to the study of U-Zr-Ti-Nb distribution in sandstone-hosted uranium ores. Spectrochimica Acta Part B: Atomic Spectroscopy. 2017 May 1;131:66-73.

[17] Kaiser J, Novotný K, Martin MZ, Hrdlička A, Malina R, Hartl M, et al. Trace elemental analysis by laser-induced breakdown spectroscopy - Biological applications. Surf Sci Rep. 2012;67(11–12):233–43.

[18] El Haddad J, Canioni L, Bousquet B. Good practices in LIBS analysis: Review and advices. Spectrochimica Acta Part B: Atomic Spectroscopy. 2014 Nov 1;101:171-82.

[19] Pořízka P, Klus J, Hrdlička A, Vrábel J, Škarková P, Prochazka D, Novotný J, Novotný K, Kaiser J. Impact of laser-induced breakdown spectroscopy data normalization on multivariate classification accuracy. Journal of Analytical Atomic Spectrometry. 2017;32(2):277-88.

[20] Pořízka P, Klus J, Prochazka D, Képeš E, Hrdlička A, Novotný J, Novotný K, Kaiser J. Laser-Induced Breakdown Spectroscopy coupled with chemometrics for the analysis of steel: The issue of spectral outliers filtering. Spectrochimica Acta Part B: Atomic Spectroscopy. 2016 Sep 1;123:114-20.

[21] Klus J, Mikysek P, Prochazka D, Pořízka P, Prochazková P, Novotný J, Trojek T, Novotný K, Slobodník M, Kaiser J. Multivariate approach to the chemical mapping of uranium in sandstone-hosted uranium ores analyzed using double pulse laser-induced breakdown spectroscopy. Spectrochimica Acta Part B: Atomic Spectroscopy. 2016 Sep 1;123:143-9.

Appendices

A. Courses, publications and presentations

ECTS Courses taken:

GS602A	Postgraduate Tutor/Demonstrator Course	5 ECTS
GS601	Intellectual Property and Commercialization	5 ECTS
CS507A	Advanced Statistics and Chemometrics	5 ECTS
FSH502	Extatic Foundation Module	5 ECTS
FSH510	EUV and X-Ray Photoionization	5 ECTS
GS607	PS Laboratory Tutoring (FSH) (Final Year)	10 ECTS
FSH509	EXTATIC Special Topics at DCU	5 ECTS
FSH508	EUV and X-Ray Metrology	5 ECTS

Publications

- Alli MB, Szwarcman D, Civitarese DS, Hayden P. Vacuum Ultraviolet Laser Induced Breakdown Spectroscopy (VUV-LIBS) with machine learning for pharmaceutical analysis. Journal of Physics: Conference Series 2019 1289, Article No. 012031
- Lu H, Varvarezos L, Alli MB, Nicolosi P, Costello JT, Hayden P. The 5d→ 6p EUV photoabsorption spectra of Pb II and Bi III: evidence of excited states. Journal of Physics B: Atomic, Molecular and Optical Physics. 2020 53 Art. No. 115001.
- Neettiyath A, Alli MB, Hayden P, Vasa NJ, Sarathi R, Costello JT. Vacuum ultraviolet laser induced breakdown spectroscopy for detecting sulphur in

thermally aged transformer insulation material. Spectrochimica Acta Part B: Atomic Spectroscopy. 2020 163 Art. No. 105730.

- Lu H, Varvarezos L, Alli MB, Nicolosi P, Costello JT, Hayden P. Reply to Comment on “The 5d→ 6p photoabsorption spectra of Pb II and Bi III: evidence of excited states” by AN Ryabtsev. Journal of Physics B: Atomic, Molecular and Optical Physics. 2021 (Accepted Manuscript).

Oral presentations

- Vacuum Ultraviolet Laser Induced Breakdown Spectroscopy (VUV-LIBS) for Pharmaceutical Analysis, M. B. Alli and P. Hayden, 9th Euro-Mediterranean Laser Induced Breakdown Spectroscopy Conference, Palazzo dei Congressi in Pisa, Italy, 11-16 June (2017)
- Vacuum Ultraviolet Laser Induced Breakdown Spectroscopy (VUV-LIBS) for Pharmaceutical Analysis, M. B. Alli, P. Hayden, J. Costello, EXTATIC Welcome Week, International Centre for Theoretical Physics Trieste, Italy, 16-20th January (2016)
- Confined Vacuum Ultraviolet (VUV) Laser Induced Breakdown Spectroscopy (LIBS) for Pharmaceutical Applications, M. B. Alli, EXTATIC Welcome Week, University of Southampton, UK, 11-15th January (2015)

Poster presentations

- Vacuum Ultraviolet Laser Induced Breakdown Spectroscopy (VUV-LIBS) With Machine Learning For Pharmaceutical Analysis, M.B. Alli, D. Szwarcman, D.S. Salles Chevitarese and P. Hayden, 24th International Conference on Spectral Line Shapes, Dublin City University, Ireland, 17-22 June (2018)
- Vacuum Ultraviolet Laser Induced Breakdown Spectroscopy (VUV-LIBS) for Pharmaceutical Analysis, M. B. Alli and P. Hayden, 9th Euro-Mediterranean

Laser Induced Breakdown Spectroscopy Conference, Palazzo dei Congressi in Pisa, Italy, 11-16 June (2017)

- Laser Induced Breakdown Spectroscopy in the Vacuum Ultraviolet spectral region (VUV-LIBS), M. B. Alli and P. Hayden, 9th International Conference on Laser-Induced Breakdown Spectroscopy, Convention Center the Majestic, Chamonix-Mont-Blanc, France, 12-16 September (2016)
- Principal Component Analysis (PCA) Applied to Vacuum Ultraviolet Laser Induced Breakdown Spectroscopy (VUV-LIBS), M. B. Alli and P. Hayden, IOP Spring Weekend 2016, Hilton Belfast, Belfast, Northern Ireland, UK, 20th February (2016)
- Vacuum Ultraviolet Laser Induced Breakdown Spectroscopy (VUV-LIBS), M.B. Alli, N. Aparna, N.J. Vasa and P. Hayden, IOP Spring Weekend 2015, Clarion Hotel, Cork, Ireland, UK, 27th February (2015)

B. Machine learning on a standard network vs an ad-hoc network

An advantage to using an industry standard network is the ease of comparing results between groups of researchers. However, transfer learning with models such as ResNet-50 (177 layers deep) which are primarily built to identify objects in traditional photos such as cats, dogs and cars contain many feature detectors that are unneeded. Each of those feature detectors uses up a vast amount of memory. over 9.6 GB of **graphics processing unit (**GPU) memory is required for the analysis of data in chapter 5. Large scale training batches such as in the data augmentation study, as seen in section 5.5.1, become impossible to run on modest hardware due to the colossal memory requirements. A simple network such as the 7-layer network used section 5.5 rarely exceeded 20 MB of memory usage on the GPU.

The 7-layer network achieved relatively comparable performance to ResNet-50 using the data from chapter 4 as shown in tables B1 and B2, the 7-layer network demonstrated better average performance than ResNet-50 using data from chapter 5 as seen in tables B3 and B4. The lower memory requirement of the 7-layer network allows for a greater overhead to be built up if more spectra is available for training, such as in the data augmentation study. The 7-layer ad-hoc network provides scalability, as more LIBS spectra are obtained, they can then be used to train networks with larger training batches.

Table B.1: Chapter 4 1vs1 testing comparing transfer learning on ResNet-50 against the ad-hoc 7-layer network.

	105 nm	85 nm	75 nm	45 nm	Average
ResNet-50	98.3	100.0	99.2	98.8	99.1
7 Layer	99.2	100.0	99.2	99.2	99.4

Table B.2: Chapter 4 multiclass testing comparing transfer learning on ResNet-50 against the ad-hoc 7-layer network.

	105 nm	85 nm	75 nm	45 nm	Average
ResNet-50	100.0	100.0	100.0	97.5	99.4
7 Layer	98.8	100.0	100.0	97.5	99.1

Table B.3: Chapter 5 1vs1 testing comparing transfer learning on ResNet-50 against the ad-hoc 7-layer network.

	105 nm	65 nm	Average
ResNet-50	85.00	95.00	90.00
7 Layer	92.9	98.8	95.8

Table B.4: Chapter 5 multiclass testing comparing transfer learning on ResNet-50 against the ad-hoc 7-layer network.

	105 nm	65 nm	Average
ResNet-50	78.8	96.9	87.8
7 Layer	91.9	95.0	93.4

C. MATLAB programs to implement classification

The code in this section shares commonalities, all of which were prototyped and run on MATLAB 2018a. Each code is supported by a framework. The framework presents the spectra to the classification technique being evaluated and records the results of the classification. Training and testing is conducted 5 times, the results are a consolidation of those 5 runs. A detailed description of the implementation is described in section 3.4.

C.1 Support vector machine classification

The code in this section carries out classification with support vector machines (SVM). The code as implemented can only be run in a 1 vs 1 arrangement. Principal component analysis (PCA) is used to dimensionally reduce the spectra down to two datapoints per spectrum as required for the input for the SVM. The SVM is trained using k-fold cross validation. Bayesian optimisation is used to tune the black box hyperparameters sigma and box. Sigma is used to designate “KernelScale”, a proprietary function that scales input data. Box is used to designate the proprietary function “BoxConstraint”, the greater the value used the higher the cost attributed to misclassification. Once the network is trained, testing is conducted comparing predicted the predicted classes to the known true classes. Finally, results are saved.

Code:

```
clear all
close all
%% Load input spectra
load('Drug_input_spectra_105nm_pos2.mat')
%% Load indexes for testing, these are the 5 randomly shuffled sets of training and
testing spectra
load('drug_indexes_105nm_pos2.mat')
%% Seed random number generator
rng(1)
%% PCA applied on the input spectra a.[1]
k=2;
z=a'*a; %Compute covariance matrix
[V,d]=sort_eig(z);% Compute e'vects and e'vals
d;
V=V(:,1:k); % Retain the first k columns of V
d=d(1:k,1:k); % Retain the kxk submatrix of e'vals
tk=a*V; % Compute the scores
holdresults=zeros(16,6);
overholdresults{1,5}=[];
%% Initialise indexes
drug1_train=train1;
drug1_test=test1;
drug2_train=train2;
drug2_test=test2;
drug3_train=train3;
drug3_test=test3;
drug4_train=train4;
drug4_test=test4;
%% j is the runs and i is the permutations, there 5 runs due to having 5 shuffles and 6
unique none repeating combinations of 1vs1 tests
for j=1:5
    for i=1:6
        close all
        if i==1
            class_1_training_index=drug1_train(j,:);
            class_1_testing_index=drug1_test(j,:);
            class_2_training_index=drug2_train(j,:);
            class_2_testing_index=drug2_test(j,:);
        elseif i==2
            class_1_training_index=drug1_train(j,:);
            class_1_testing_index=drug1_test(j,:);
            class_2_training_index=drug3_train(j,:);
            class_2_testing_index=drug3_test(j,:);
        elseif i==3
            class_1_training_index=drug1_train(j,:);
            class_1_testing_index=drug1_test(j,:);
            class_2_training_index=drug4_train(j,:);
            class_2_testing_index=drug4_test(j,:);
```

```
        elseif i==4
          class_1_training_index=drug2_train(j,:);
          class_1_testing_index=drug2_test(j,:);
          class_2_training_index=drug3_train(j,:);
          class_2_testing_index=drug3_test(j,:);
        elseif i==5
          class_1_training_index=drug2_train(j,:);
          class_1_testing_index=drug2_test(j,:);
          class_2_training_index=drug4_train(j,:);
          class_2_testing_index=drug4_test(j,:);
        elseif i==6
          class_1_training_index=drug3_train(j,:);
          class_1_testing_index=drug3_test(j,:);
          class_2_training_index=drug4_train(j,:);
          class_2_testing_index=drug4_test(j,:);
          %% Here the training and testing indexes are set
        end
        class1=tk(class_1_training_index,:);
        class2=tk(class_2_training_index,:);
        cdata = [class1;class2];
        %% Class labelling
        grp = ones(32,1);
        grp(17:32,1) =-1;
        c = cvpartition(32,'KFold',4);% 32 denotes the number of spectra, 4 is how k
folded it is, set 4 to 32 to use leave one out cross validation
        %% Train network
        % below the black box variables to be tuned by Bayesian optimisation are
designated [ 2]
sigma =optimizableVariable('sigma',[1e5,1e5],'Transform','log');
box = optimizableVariable('box',[1e5,1e5],'Transform','log');

        minfn = @(z)kfoldLoss(fitcsvm(cdata,grp,'CVPartition',c,...
          'KernelFunction','rbf','BoxConstraint',z.box,...
          'KernelScale',z.sigma));
        results = bayesopt(minfn,[sigma,box],'IsObjectiveDeterministic',true,...
          'AcquisitionFunctionName','expected-improvement-plus','Verbose', 0, 'PlotFcn',
[]);
        z(1) = results.XAtMinObjective.sigma;
        z(2) = results.XAtMinObjective.box;
        SVMModel = fitcsvm(cdata,grp,'KernelFunction','rbf',...
          'KernelScale',z(1),'BoxConstraint',z(2));
        %% Load testing data
        grnobj = tk(class_1_testing_index,:);
        redobj = tk(class_2_testing_index,:);
        newData = [grnobj ;redobj];
        grpData = ones(16,1);
        grpData(9:16) = -1;
        v = predict(SVMModel,newData);%use model to predict the classes spectra
belong to
        mydiff = (v == grpData); % Classified correctly
```

```
        holdresults(:,i)=mydiff;
        if i==6
            overholdresults{1,j}=holdresults;%Save and consolidate results in a single
variable
        else
        end
    end
end
% Results are saved

save('results_105nm_pos1_rbf.mat','overholdresults');
%References
%[1] Gemperline P, editor. Practical guide to chemometrics. CRC press; 2006 Apr 16.
%[2] MathWorks. Support Vector Machines for Binary
Classification[Internet].2017[Cited 13/02/1018]. Available
from:https://www.mathworks.com/help/stats/support-vector-machines-for-binary-
classification.html#bsr5o1q
%[3] Hastie T, Tibshirani R, Friedman J. The elements of statistical learning: data
mining, inference, and prediction. Springer Science & Business Media; 2009 Aug 26.
%[4] Christianini, N, and J Shawe-Taylor. An Introduction to Support Vector Machines
and Other Kernel-Based Learning Methods. Cambridge, UK: Cambridge University
Press, 2000.
%[5] Fan RE, Chen PH, Lin CJ. Working set selection using second order information
for training support vector machines. Journal of machine learning research.
2005;6(Dec):1889-918.
%[6] Kecman V, Huang TM, Vogt M. Iterative single data algorithm for training kernel
machines from huge data sets: Theory and performance. In Support vector machines:
Theory and Applications 2005 Jun 21 (pp. 255-274). Springer, Berlin, Heidelberg.
```

C.2 Principal component analysis: multiclass classification

The code in this section carries out multiclass classification with PCA. PCA is used to dimensionally reduce the spectra down to three datapoints per spectrum. Training spectra are used to generate an ellipsoid for each class. For testing each spectrum is checked to see if it belongs to the appropriate class and no other class. Finally, results are saved.

Code:

```
clear all
close all
%  tic
%% Load in data
load('Drug_input_spectra_65nm_pos2.mat')
%% Load indexes for testing
load('drug_indexes_65nm_pos2.mat')
%% Preallocation of variables
holdsamples=zeros(1,32);
holdpermutations{1,5}=[];
Unique_Neurons_populated{1,5}=[];
%% PCA applied on the input spectra a.[1]
k=3;
z=a'*a; % compute covariance matrix
[V,d]=sort_eig(z);% compute e'vects and e'vals
d;
 % select the number of factors
V=V(:,1:k); % retain the first k columns of V
d=d(1:k,1:k); % retain the kxk submatrix of e'vals
tk=a*V;
a=tk;
a=double(a);
%% Initialise indexes
drug1_train=train1;
drug1_test=test1;
drug2_train=train2;
drug2_test=test2;
drug3_train=train3;
 drug3_test=test3;
 drug4_train=train4;
drug4_test=test4;
        for runs=1:5
        class_1_training_index=drug1_train(runs,:);
        class_1_testing_index=drug1_test(runs,:);
        class_2_training_index=drug2_train(runs,:);
        class_2_testing_index=drug2_test(runs,:);


        class_3_training_index=drug3_train(runs,:);
        class_3_testing_index=drug3_test(runs,:);
        class_4_training_index=drug4_train(runs,:);
        class_4_testing_index=drug4_test(runs,:);
      grnobj = a(class_1_testing_index,:);
redobj = a(class_2_testing_index,:);
bluob=a(class_3_testing_index,:);
yelobj=a(class_4_testing_index,:);
    newData = [grnobj;redobj;bluob;yelobj;];
for testspectra=1:32
```

```
    class1=a(class_1_training_index,:);
    class2=a(class_2_training_index,:);
    class3=a(class_3_training_index,:);
    class4=a(class_4_training_index,:);
    cdata = [class1;class2;class3;class4];
    cdata(65,:)= newData(testspectra,:);
    punk= newData(testspectra,:);
    punk=double(punk);
    %% Ellipsoids are generated and checked for points inside their volume [2].

    %%     Ellipse 1

    mpx=(sum(class1(:,1)))/16;
    mpy=(sum(class1(:,2)))/16;
    mpz=(sum(class1(:,3)))/16;
    sigx=std(class1(:,1))*3;
    sigy=std(class1(:,2))*3;
    sigz=std(class1(:,3))*3;
    [x, y, z] = ellipsoid(mpx,mpy,mpz,sigx,sigy,sigz,100);
    big=length(x);
    verybig=big*big;
% Variables are reshaped
    XS = reshape(x,[verybig,1]);
    YS = reshape(y,[verybig,1]);
    ZS = reshape(z,[verybig,1]);
    elipse(:,1)=XS;
    elipse(:,2)=YS;
    elipse(:,3)=ZS;
    % in1 will be 1 if the point is in the volume of the ellipse and 0 if it is outside the
ellipse
    in1 = inhull(punk,elipse);
    %%     Ellipse 2
    mpx=(sum(class2(:,1)))/16;
    mpy=(sum(class2(:,2)))/16;
    mpz=(sum(class2(:,3)))/16;

    sigx=std(class2(:,1))*3;
    sigy=std(class2(:,2))*3;
    sigz=std(class2(:,3))*3;
    [x, y, z] = ellipsoid(mpx,mpy,mpz,sigx,sigy,sigz,100);
    big=length(x);
    verybig=big*big;
    % Variables are reshaped
    XS = reshape(x,[verybig,1]);
    YS = reshape(y,[verybig,1]);
    ZS = reshape(z,[verybig,1]);
    elipse(:,1)=XS;
    elipse(:,2)=YS;
    elipse(:,3)=ZS;
    % in1 will be 1 if the point is in the volume of the ellipse and 0 if it is outside the
ellipse
```

```
   in2 = inhull(punk,elipse);
   %% Ellipse 3
   mpx=(sum(class3(:,1)))/16;
   mpy=(sum(class3(:,2)))/16;
   mpz=(sum(class3(:,3)))/16;

   sigx=std(class3(:,1))*3;
   sigy=std(class3(:,2))*3;
   sigz=std(class3(:,3))*3;
   [x, y, z] = ellipsoid(mpx,mpy,mpz,sigx,sigy,sigz,100);
   big=length(x);
   verybig=big*big;
   % Variables are reshaped
   XS = reshape(x,[verybig,1]);
   YS = reshape(y,[verybig,1]);
   ZS = reshape(z,[verybig,1]);
   elipse(:,1)=XS;
   elipse(:,2)=YS;
   elipse(:,3)=ZS;
   % in1 will be 1 if the point is in the volume of the ellipse and 0 if it is outside the
ellipse
   in3 = inhull(punk,elipse);
   %% Ellipse 4
   mpx=(sum(class4(:,1)))/16;
   mpy=(sum(class4(:,2)))/16;
   mpz=(sum(class4(:,3)))/16;
   sigx=std(class4(:,1))*3;
   sigy=std(class4(:,2))*3;
   sigz=std(class4(:,3))*3;
   [x, y, z] = ellipsoid(mpx,mpy,mpz,sigx,sigy,sigz,100);
   big=length(x);
   verybig=big*big;
   % Variables are reshaped
   XS = reshape(x,[verybig,1]);
   YS = reshape(y,[verybig,1]);
   ZS = reshape(z,[verybig,1]);
   elipse(:,1)=XS;
   elipse(:,2)=YS;
   elipse(:,3)=ZS;
   % in1 will be 1 if the point is in the volume of the ellipse and 0 if it is outside the
ellipse
   in4 = inhull(punk,elipse);
   %% Sets are isolated, intersect is used
   % to determine if there are common elements between 2 sets
   if testspectra<=8  && in1==1 && in2==0 && in3==0 && in4==0
      holdsamples(testspectra)=1;
   elseif testspectra > 8 && testspectra<=16  && in1==0 && in2==1 && in3==0 &&
in4==0
      holdsamples(testspectra)=1;
```

```
   elseif testspectra > 16 && testspectra<=24  && in1==0 && in2==0 && in3==1 &&
in4==0
       holdsamples(testspectra)=1;
   elseif testspectra > 24 && testspectra<=32  && in1==0 && in2==0 && in3==0 &&
in4==1
       holdsamples(testspectra)=1;
   else
       holdsamples(testspectra)=0;
   end
   if testspectra==32
       holdpermutations{1,runs}=holdsamples;
   else
   end
end
end
% Results are saved
save('resultsDrugs_65nm_pos2_milticlass_ PCA.mat','holdpermutations')
%%References
%[1] Gemperline P, editor. Practical guide to chemometrics. CRC press; 2006 Apr 16.
%[2] D'Errico J. Inhull [Internet]. 2012[cited 03/04/2016]. Available
from:https://www.mathworks.com/matlabcentral/fileexchange/10226-inhull
```

C.3 Principal component analysis: 1 vs 1 classification

The code in this section carries out 1 vs 1 classification with PCA. PCA is used to dimensionally reduce the spectra down to three datapoints per spectrum. Training spectra are used to generate an ellipsoid for each class. For testing each spectrum is checked to see if it belongs to the appropriate class and no other class. Finally, results are saved.

Code:

```
clear all
close all
%% Load in data
load('Drug_input_spectra_65nm_pos2.mat')
%% Load indexes for testing
load('drug_indexes_65nm_pos2.mat')
%% Preallocation of variables
No_common_elements=zeros(1,2);
holdsamples=zeros(16,6);
```

```
holdpermutations{1,5}=[];
Unique_Neurons_populated{1,3}=[];
%% PCA applied on the input spectra a.[1]
k=3;
z=a'*a; % compute covariance matrix
[V,d]=sort_eig(z);% compute e'vects and e'vals
d;
V=V(:,1:k); % retain the first k columns of V
d=d(1:k,1:k); % retain the kxk submatrix of e'vals
tk=a*V;
a=tk;
a=double(a);
%% Initialise indexes
drug1_train=train1;
drug1_test=test1;
drug2_train=train2;
drug2_test=test2;
drug3_train=train3;
drug3_test=test3;
drug4_train=train4;
drug4_test=test4;
for runs=1:5
    for permutations=1:6
        if permutations==1
            class_1_training_index=drug1_train(runs,:);
            class_1_testing_index=drug1_test(runs,:);
            class_2_training_index=drug2_train(runs,:);
            class_2_testing_index=drug2_test(runs,:);
        elseif permutations==2
            class_1_training_index=drug1_train(runs,:);
            class_1_testing_index=drug1_test(runs,:);
            class_2_training_index=drug3_train(runs,:);
            class_2_testing_index=drug3_test(runs,:);
        elseif permutations==3
            class_1_training_index=drug1_train(runs,:);
            class_1_testing_index=drug1_test(runs,:);
            class_2_training_index=drug4_train(runs,:);
            class_2_testing_index=drug4_test(runs,:);
        elseif permutations==4
            class_1_training_index=drug2_train(runs,:);
            class_1_testing_index=drug2_test(runs,:);
            class_2_training_index=drug3_train(runs,:);
            class_2_testing_index=drug3_test(runs,:);
        elseif permutations==5
            class_1_training_index=drug2_train(runs,:);
            class_1_testing_index=drug2_test(runs,:);
            class_2_training_index=drug4_train(runs,:);
            class_2_testing_index=drug4_test(runs,:);
        elseif permutations==6
            class_1_training_index=drug3_train(runs,:);
```

```
            class_1_testing_index=drug3_test(runs,:);
            class_2_training_index=drug4_train(runs,:);
            class_2_testing_index=drug4_test(runs,:);
        end
        grnobj = a(class_1_testing_index,:);
        redobj = a(class_2_testing_index,:);
        newData = [grnobj ;redobj];
        for testspectra=1:16
            class1=a(class_1_training_index,:);
            class2=a(class_2_training_index,:);
            cdata = [class1;class2];
            cdata(33,:)= newData(testspectra,:);
            punk= newData(testspectra,:);
            punk=double(punk);
            %%    Ellipse 1
            mpx=(sum(class1(:,1)))/16;
            mpy=(sum(class1(:,2)))/16;
            mpz=(sum(class1(:,3)))/16;
            sigx=std(class1(:,1))*3;
            sigy=std(class1(:,2))*3;
            sigz=std(class1(:,3))*3;
            [x, y, z] = ellipsoid(mpx,mpy,mpz,sigx,sigy,sigz,100);
            big=length(x);
            verybig=big*big;
            % Variables are reshaped
            XS = reshape(x,[verybig,1]);
            YS = reshape(y,[verybig,1]);
            ZS = reshape(z,[verybig,1]);
            elipse(:,1)=XS;
            elipse(:,2)=YS;
            elipse(:,3)=ZS;
            % in1 will be 1 if the point is in the volume of the ellipse and 0 if it is outside the ellipse
            in1 = inhull(punk,elipse);
            %%    Ellipse 2
            mpx=(sum(class2(:,1)))/16;
            mpy=(sum(class2(:,2)))/16;
            mpz=(sum(class2(:,3)))/16;
            sigx=std(class2(:,1))*3;
            sigy=std(class2(:,2))*3;
            sigz=std(class2(:,3))*3;
            [x, y, z] = ellipsoid(mpx,mpy,mpz,sigx,sigy,sigz,100);
            big=length(x);
            verybig=big*big;
            % Variables are reshaped
            XS = reshape(x,[verybig,1]);
            YS = reshape(y,[verybig,1]);
            ZS = reshape(z,[verybig,1]);
            elipse(:,1)=XS;
            elipse(:,2)=YS;
```

```
      elipse(:,3)=ZS;
      % in2 will be 1 if the point is in the volume of the ellipse and 0 if it is outside
the ellipse
      in2 = inhull(punk,elipse);
      common_elements{1,1}=in1;
      common_elements{1,2}=in2;
      for i=1:2
        No_common_elements(1,i)=common_elements{1,i};
      end
      if testspectra<=8 &&  No_common_elements(1,1)==1 &&
No_common_elements(1,2)==0
        holdsamples(testspectra,permutations)=1;
      elseif testspectra > 8 &&  No_common_elements(1,2)==1 &&
No_common_elements(1,1)==0
        holdsamples(testspectra,permutations)=1;
      else
        holdsamples(testspectra,permutations)=0;
      end
      if permutations==6 && testspectra==16
        holdpermutations{1,runs}=holdsamples;
      else
      end
    end
  end
end
% Results are saved
save('results_1_vs1_55nm_pos2_PCA.mat','holdpermutations')
 %References
%[1] Gemperline P, editor. Practical guide to chemometrics. CRC press; 2006 Apr 16.
%[2] D'Errico J. Inhull [Internet]. 2012[cited 03/04/2016]. Available
from:https://www.mathworks.com/matlabcentral/fileexchange/10226-inhull
```

C.4 Self-organizing maps: 4x4 multiclass classification

The code in this section carries out multiclass classification with self-organizing maps (SOM). Training spectra are used to dope the network and are loaded simultaneously with a single test spectrum at a time. For testing, each test spectrum is checked to see if it belongs to the appropriate class and no other class. This is facilitated by the intersect MATLAB function. Each class represents a set, if a spectrum belongs only to a single set, then it will not intersect with any other set. Finally, results are saved.

Code:

```
clear all
close all
%% Load in data
load('Drug_input_spectra_65nm_pos2.mat')
%% Load indexes for testing
load('drug_indexes_65nm_pos2.mat')
%% Preallocation of variables
No_common_elements=zeros(1,10);
hold_class_info=zeros(1,65);
Neurons_populated{1,5}=[];
holdsamples=zeros(1,32);
holdpermutations{1,5}=[];
Unique_Neurons_populated{1,5}=[];
%% Seed random number generator
rng(1)
%% Initialise indexes
drug1_train=train1;
drug1_test=test1;
drug2_train=train2;
drug2_test=test2;
drug3_train=train3;
drug3_test=test3;
drug4_train=train4;
drug4_test=test4;
for runs=1:5
   class_1_training_index=drug1_train(runs,:);
   class_1_testing_index=drug1_test(runs,:);
   class_2_training_index=drug2_train(runs,:);
   class_2_testing_index=drug2_test(runs,:);
   class_3_training_index=drug3_train(runs,:);
   class_3_testing_index=drug3_test(runs,:);
   class_4_training_index=drug4_train(runs,:);
   class_4_testing_index=drug4_test(runs,:);
   grnobj = a(class_1_testing_index,:);
   redobj = a(class_2_testing_index,:);
   bluob=a(class_3_testing_index,:);
   yelobj=a(class_4_testing_index,:);
   newData = [grnobj;redobj;bluob;yelobj;];
   for testspectra=1:32

      class1=a(class_1_training_index,:);
      class2=a(class_2_training_index,:);
      class3=a(class_3_training_index,:);
      class4=a(class_4_training_index,:);
      cdata = [class1;class2;class3;class4];
      cdata(65,:)= newData(testspectra,:);
      cdata =transpose(cdata);
      net = selforgmap([4 4]); % here the number and layout of neurons is designated
```

```
    net.trainParam.showWindow = false;
    net = train(net,cdata);
    y = net(cdata);
    number_of_samples=size(y,2);
    number_of_classes=size(y,1);
    %%  The following code is used to assess which neurons are hit by which samples
    for i=1:number_of_samples
      for k=1:number_of_classes
        if y(k,i)==1
          hold_class_info(1,i)=k;
        else
        end
      end
    end
    index_neurons=1;
    for i=1:16:64
      Neurons_populated{index_neurons}=hold_class_info(1,(i:i+15));
      index_neurons=index_neurons+1;
    end
    Neurons_populated{5} =hold_class_info(1,65);
    for i=1:5
      holdme=Neurons_populated{1,i};
      Unique_Neurons_populated{1,i}=unique(holdme);
    end
    nntraintool('close')
    %% Sets are isolated, intersect is used
    % to determine if there are common elements between 2 sets
set1=Unique_Neurons_populated{1,1};
    set2=Unique_Neurons_populated{1,2};
    set3=Unique_Neurons_populated{1,3};
    set4=Unique_Neurons_populated{1,4};
    set5=Unique_Neurons_populated{1,5};
    % e.g., of this test below, all repeating comparisons removed i.e., 1vs2 = 2vs1,
    % and self-comparisons removed i.e., 1vs1 2vs2 because we know these sets are
    % identical and therefore will intersect
    common_elements{1,1}=intersect(set1,set2);
    common_elements{1,2}=intersect(set1,set3);
    common_elements{1,3}=intersect(set1,set4);
    common_elements{1,4}=intersect(set2,set3);
    common_elements{1,5}=intersect(set2,set4);
    common_elements{1,6}=intersect(set3,set4);
    common_elements{1,7}=intersect(set1,set5);
    common_elements{1,8}=intersect(set2,set5);
    common_elements{1,9}=intersect(set3,set5);
    common_elements{1,10}=intersect(set4,set5);
    %If there are no common elements an empty matrix is returned,
    % We can use the isempty function to check for an empty matrix,
    % it returns a one if true, and zero if false.
    for i=1:10
      No_common_elements(1,i)=isempty(common_elements{1,i});
```

```
        end

check_for_no_crossover1=No_common_elements(1,1)+No_common_elements(1,2)+N
o_common_elements(1,3);

check_for_no_crossover2=No_common_elements(1,1)+No_common_elements(1,4)+N
o_common_elements(1,5);

check_for_no_crossover3=No_common_elements(1,2)+No_common_elements(1,4)+N
o_common_elements(1,6);

check_for_no_crossover4=No_common_elements(1,3)+No_common_elements(1,5)+N
o_common_elements(1,6);
        if testspectra<=8  && No_common_elements(1,7)==0 &&
check_for_no_crossover1==3
            holdsamples(testspectra)=1;
        elseif testspectra > 8 && testspectra<=16  && No_common_elements(1,8)==0
&& check_for_no_crossover2==3
            holdsamples(testspectra)=1;
        elseif testspectra > 16 && testspectra<=24  && No_common_elements(1,9)==0
&& check_for_no_crossover3==3
            holdsamples(testspectra)=1;
        elseif testspectra > 24 && testspectra<=32  && No_common_elements(1,10)==0
&& check_for_no_crossover4==3
            holdsamples(testspectra)=1;
        else
            holdsamples(testspectra)=0;
        end
        if testspectra==32
            holdpermutations{1,runs}=holdsamples;
        else
        end
    end
end
% Results are saved
save('resultsDrugs_65nm_pos2_milticlass_ SOM_4x4.mat','holdpermutations')
```

C.5 Self-organizing maps 4x4: 1 vs 1 classification

The code in this section carries out 1 vs 1 classification with SOMs. Training spectra are used to dope the network and are loaded simultaneously with a single test spectrum at a time. For testing, each test spectrum is checked to see if it belongs to the appropriate class and no other class. This is facilitated by the intersect MATLAB function. Each class represents a set, if a spectrum belongs only to a single set, then it will not intersect with any other set. Finally, results are saved.

Code:

```
clear all
close all
%% Load in data
load('Drug_input_spectra_65nm_pos2.mat')
%% Load indexes for testing
load('drug_indexes_65nm_pos2.mat')
%% Preallocation of variables
No_common_elements=zeros(1,2);
hold_class_info=zeros(1,33);
Neurons_populated{1,3}=[];
holdsamples=zeros(16,6);
holdpermutations{1,5}=[];
Unique_Neurons_populated{1,3}=[];
%% Seed random number generator
rng(1)
%% Initialise indexes
drug1_train=train1;
drug1_test=test1;
drug2_train=train2;
drug2_test=test2;
drug3_train=train3;
drug3_test=test3;
drug4_train=train4;
drug4_test=test4;
for runs=1:5
    for permutations=1:6
        if permutations==1
            class_1_training_index=drug1_train(runs,:);
            class_1_testing_index=drug1_test(runs,:);
            class_2_training_index=drug2_train(runs,:);
            class_2_testing_index=drug2_test(runs,:);
        elseif permutations==2
```

```
        class_1_training_index=drug1_train(runs,:);
        class_1_testing_index=drug1_test(runs,:);
        class_2_training_index=drug3_train(runs,:);
        class_2_testing_index=drug3_test(runs,:);
    elseif permutations==3
        class_1_training_index=drug1_train(runs,:);
        class_1_testing_index=drug1_test(runs,:);
        class_2_training_index=drug4_train(runs,:);
        class_2_testing_index=drug4_test(runs,:);
    elseif permutations==4
        class_1_training_index=drug2_train(runs,:);
        class_1_testing_index=drug2_test(runs,:);
        class_2_training_index=drug3_train(runs,:);
        class_2_testing_index=drug3_test(runs,:);
    elseif permutations==5
        class_1_training_index=drug2_train(runs,:);
        class_1_testing_index=drug2_test(runs,:);
        class_2_training_index=drug4_train(runs,:);
        class_2_testing_index=drug4_test(runs,:);
    elseif permutations==6
        class_1_training_index=drug3_train(runs,:);
        class_1_testing_index=drug3_test(runs,:);
        class_2_training_index=drug4_train(runs,:);
        class_2_testing_index=drug4_test(runs,:);
    end
    grnobj = a(class_1_testing_index,:);
    redobj = a(class_2_testing_index,:);
    newData = [grnobj ;redobj];
    for testspectra=1:16
        class1=a(class_1_training_index,:);
        class2=a(class_2_training_index,:);
        cdata = [class1;class2];
        cdata(33,:)= newData(testspectra,:);
        cdata =transpose(cdata);
        net = selforgmap([4 4]); % here the number and layout of neurons is designated
        net.trainParam.showWindow = false;
        net = train(net,cdata);
        y = net(cdata);
        number_of_samples=size(y,2);
        number_of_classes=size(y,1);
        %% The following code is used to assess which neurons are hit by which samples
        for i=1:number_of_samples
            for k=1:number_of_classes
                if y(k,i)==1
                    hold_class_info(1,i)=k;
                else
                end
            end
        end
```

```
         index_neurons=1;
         for i=1:16:32
            Neurons_populated{index_neurons}=hold_class_info(1,(i:i+15));
            index_neurons=index_neurons+1;
         end
         Neurons_populated{3} =hold_class_info(1,33);
         for i=1:3
            holdme=Neurons_populated{1,i};
            Unique_Neurons_populated{1,i}=unique(holdme);
         end
         nntraintool('close')
         %% Sets are isolated, intersect is used
         % to determine if there are common elements between 2 sets
set1=Unique_Neurons_populated{1,1};
         set2=Unique_Neurons_populated{1,2};
         set3=Unique_Neurons_populated{1,3};
         % e.g., of this test below, all repeating comparisons removed i.e., 1vs2 = 2vs1,
      % and self-comparisons removed i.e., 1vs1 2vs2 because we know these sets are
      % identical and therefore will intersect
common_elements{1,1}=intersect(set1,set3);
         common_elements{1,2}=intersect(set2,set3);
         % If there are no common elements an empty matrix is returned
         % we can use the isempty function to check for an empty matrix
         % it returns a one if true, and zero if false
         for i=1:2
            No_common_elements(1,i)=isempty(common_elements{1,i});
         end
         if testspectra<=8 &&  No_common_elements(1,1)==0 &&
No_common_elements(1,2)==1
            holdsamples(testspectra,permutations)=1;
         elseif testspectra > 8 &&  No_common_elements(1,2)==0 &&
No_common_elements(1,1)==1
            holdsamples(testspectra,permutations)=1;
         else
            holdsamples(testspectra,permutations)=0;
         end
         if permutations==6 && testspectra==16
            holdpermutations{1,runs}=holdsamples;
         else
         end
      end

   end
end
save('results_1_vs1_65nm_pos2_SOM_4x4.mat','holdpermutations')
% Results are saved
```

C.6 Competitive learning: multiclass classification

The code in this section carries out multiclass classification with competitive learning (COMP). Training spectra are used to dope the network and are loaded simultaneously with a single test spectrum at a time. For testing, each test spectrum is checked to see if it belongs to the appropriate class and no other class. This is facilitated by the intersect MATLAB function. Each class represents a set, if a spectrum belongs only to a single set, then it will not intersect with any other set. Finally, results are saved.

Code:

```
clear all
close all
%% Load in data
load('Drug_input_spectra_65nm_pos2.mat')
%% Load indexes for testing
load('drug_indexes_65nm_pos2.mat')
%% Preallocation of variables
No_common_elements=zeros(1,10);
hold_class_info=zeros(1,65);
Neurons_populated{1,5}=[];
holdsamples=zeros(1,32);
holdpermutations{1,5}=[];
Unique_Neurons_populated{1,5}=[];
%% Seed random number generator
rng(1)
%% Initialise indexes
drug1_train=train1;
drug1_test=test1;
drug2_train=train2;
drug2_test=test2;
drug3_train=train3;
drug3_test=test3;
drug4_train=train4;
drug4_test=test4;
for runs=1:5
    class_1_training_index=drug1_train(runs,:);
    class_1_testing_index=drug1_test(runs,:);
    class_2_training_index=drug2_train(runs,:);
    class_2_testing_index=drug2_test(runs,:);
    class_3_training_index=drug3_train(runs,:);
    class_3_testing_index=drug3_test(runs,:);
    class_4_training_index=drug4_train(runs,:);
```

```
    class_4_testing_index=drug4_test(runs,:);
    grnobj = a(class_1_testing_index,:);
    redobj = a(class_2_testing_index,:);
    bluob=a(class_3_testing_index,:);
    yelobj=a(class_4_testing_index,:);
    newData = [grnobj;redobj;bluob;yelobj;];
    for testspectra=1:32
        class1=a(class_1_training_index,:);
        class2=a(class_2_training_index,:);
        class3=a(class_3_training_index,:);
        class4=a(class_4_training_index,:);
        cdata = [class1;class2;class3;class4];
        cdata(65,:)= newData(testspectra,:);
        cdata =transpose(cdata);
        net = competlayer(4); % here the number of neurons is designated, this is set equal
to the number of classes.
        net.trainParam.showWindow = false;
        net = train(net,cdata);
        y = net(cdata);
        number_of_samples=size(y,2);
        number_of_classes=size(y,1);
        %% The following code is used to assess which neurons are hit by which samples
        for i=1:number_of_samples
            for k=1:number_of_classes
                if y(k,i)==1
                    hold_class_info(1,i)=k;
                else
                end
            end
        end
        index_neurons=1;
        for i=1:16:64
            Neurons_populated{index_neurons}=hold_class_info(1,(i:i+15));
            index_neurons=index_neurons+1;
        end
        Neurons_populated{5} =hold_class_info(1,65);
        for i=1:5
            holdme=Neurons_populated{1,i};
            Unique_Neurons_populated{1,i}=unique(holdme);
        end
        nntraintool('close')
        %% Sets are isolated, intersect is used
    % to determine if there are common elements between 2 sets
set1=Unique_Neurons_populated{1,1};
        set2=Unique_Neurons_populated{1,2};
        set3=Unique_Neurons_populated{1,3};
        set4=Unique_Neurons_populated{1,4};
        set5=Unique_Neurons_populated{1,5};
        % e.g. of this test below, all repeating comparisons removed i.e. 1vs2 = 2vs1,
        % and self-comparisons removed i.e. 1vs1 2vs2 because we know these sets are
```

```
        % identical and therefore will intersect
        common_elements{1,1}=intersect(set1,set2);
        common_elements{1,2}=intersect(set1,set3);
        common_elements{1,3}=intersect(set1,set4);
        common_elements{1,4}=intersect(set2,set3);
        common_elements{1,5}=intersect(set2,set4);
        common_elements{1,6}=intersect(set3,set4);
        common_elements{1,7}=intersect(set1,set5);
        common_elements{1,8}=intersect(set2,set5);
        common_elements{1,9}=intersect(set3,set5);
        common_elements{1,10}=intersect(set4,set5);
        %If there are no common elements an empty matrix is returned
        % we can use the isempty function to check for an empty matrix
        % it returns a one if true, and zero if false
        for i=1:10
            No_common_elements(1,i)=isempty(common_elements{1,i});
        end
check_for_no_crossover1=No_common_elements(1,1)+No_common_elements(1,2)+N
o_common_elements(1,3);
check_for_no_crossover2=No_common_elements(1,1)+No_common_elements(1,4)+N
o_common_elements(1,5);
check_for_no_crossover3=No_common_elements(1,2)+No_common_elements(1,4)+N
o_common_elements(1,6);
check_for_no_crossover4=No_common_elements(1,3)+No_common_elements(1,5)+N
o_common_elements(1,6);
        if testspectra<=8  && No_common_elements(1,7)==0 &&
check_for_no_crossover1==3
            holdsamples(testspectra)=1;
        elseif testspectra > 8 && testspectra<=16  && No_common_elements(1,8)==0
&& check_for_no_crossover2==3
            holdsamples(testspectra)=1;
        elseif testspectra > 16 && testspectra<=24  && No_common_elements(1,9)==0
&& check_for_no_crossover3==3
            holdsamples(testspectra)=1;
        elseif testspectra > 24 && testspectra<=32  && No_common_elements(1,10)==0
&& check_for_no_crossover4==3
            holdsamples(testspectra)=1;
        else
            holdsamples(testspectra)=0;
        end
        if testspectra==32
            holdpermutations{1,runs}=holdsamples;
        else
        end
    end
end
% Results are saved
save('comp_rstsDrugs_65nm_pos2_mlticlas.mat','holdpermutations')
```

C.7 Competitive learning: 1 vs 1 classification

The code in this section carries out 1 vs 1 classification with COMP. Training spectra are used to dope the network and are loaded simultaneously with a single test spectrum at a time. For testing, each test spectrum is checked to see if it belongs to the appropriate class and no other class. This is facilitated by the intersect MATLAB function. Each class represents a set, if a spectrum belongs only to a single set, then it will not intersect with any other set. Finally, results are saved.

Code:

```
clear all
close all
%% Load in data
load('Drug_input_spectra_65nm_pos2.mat')
%% Load indexes for testing
load('drug_indexes_65nm_pos2.mat')
%% Preallocation of variables
No_common_elements=zeros(1,2);
hold_class_info=zeros(1,33);
Neurons_populated{1,3}=[];
holdsamples=zeros(16,6);
holdpermutations{1,5}=[];
Unique_Neurons_populated{1,3}=[];
%% Seed random number generator
rng(1)
%% Initialise indexes
drug1_train=train1;
drug1_test=test1;
drug2_train=train2;
drug2_test=test2;
drug3_train=train3;
drug3_test=test3;
drug4_train=train4;
drug4_test=test4;
for runs=1:5
    for permutations=1:6
        if permutations==1
            class_1_training_index=drug1_train(runs,:);
            class_1_testing_index=drug1_test(runs,:);
            class_2_training_index=drug2_train(runs,:);
            class_2_testing_index=drug2_test(runs,:);
```

```
        elseif permutations==2
            class_1_training_index=drug1_train(runs,:);
            class_1_testing_index=drug1_test(runs,:);
            class_2_training_index=drug3_train(runs,:);
            class_2_testing_index=drug3_test(runs,:);
        elseif permutations==3
            class_1_training_index=drug1_train(runs,:);
            class_1_testing_index=drug1_test(runs,:);
            class_2_training_index=drug4_train(runs,:);
            class_2_testing_index=drug4_test(runs,:);
        elseif permutations==4
            class_1_training_index=drug2_train(runs,:);
            class_1_testing_index=drug2_test(runs,:);
            class_2_training_index=drug3_train(runs,:);
            class_2_testing_index=drug3_test(runs,:);
        elseif permutations==5
            class_1_training_index=drug2_train(runs,:);
            class_1_testing_index=drug2_test(runs,:);
            class_2_training_index=drug4_train(runs,:);
            class_2_testing_index=drug4_test(runs,:);
        elseif permutations==6
            class_1_training_index=drug3_train(runs,:);
            class_1_testing_index=drug3_test(runs,:);
            class_2_training_index=drug4_train(runs,:);
            class_2_testing_index=drug4_test(runs,:);
        end
        grnobj = a(class_1_testing_index,:);
        redobj = a(class_2_testing_index,:);
        newData = [grnobj ;redobj];
        for testspectra=1:16
            class1=a(class_1_training_index,:);
            class2=a(class_2_training_index,:);
            cdata = [class1;class2];
            cdata(33,:)= newData(testspectra,:);
            cdata =transpose(cdata);
            net = competlayer(2);% here the number of neurons is designated, this is set
equal to the number of classes.
            net.trainParam.showWindow = false;
            net = train(net,cdata);
            y = net(cdata);
            number_of_samples=size(y,2);
            number_of_classes=size(y,1);
            %% The following code is used to assess which neurons are hit by which
samples
            for i=1:number_of_samples
                for k=1:number_of_classes
                    if y(k,i)==1
                        hold_class_info(1,i)=k;
                    else
                    end
```

```
                end
            end
            index_neurons=1;
            for i=1:16:32
                Neurons_populated{index_neurons}=hold_class_info(1,(i:i+15));
                index_neurons=index_neurons+1;
            end
            Neurons_populated{3} =hold_class_info(1,33);
            for i=1:3
                holdme=Neurons_populated{1,i};
                Unique_Neurons_populated{1,i}=unique(holdme);
            end
            nntraintool('close')
            %% Sets are isolated, intersect is used
    % to determine if there are common elements between 2 sets
            set1=Unique_Neurons_populated{1,1};
            set2=Unique_Neurons_populated{1,2};
            set3=Unique_Neurons_populated{1,3};
            % e.g. of this test below, all repeating comparisons removed i.e. 1vs2 = 2vs1,
        % and self-comparisons removed i.e. 1vs1 2vs2 because we know these sets are
        % identical and therefore will intersect
            common_elements{1,1}=intersect(set1,set3);
            common_elements{1,2}=intersect(set2,set3);
            %If there are no common elements an empty matrix is returned
            % we can use the isempty function to check for an empty matrix
            % it returns a one if true, and zero if false
            for i=1:2
                No_common_elements(1,i)=isempty(common_elements{1,i});
            end
            if testspectra<=8 &&  No_common_elements(1,1)==0 &&
No_common_elements(1,2)==1
                holdsamples(testspectra,permutations)=1;
            elseif testspectra > 8 &&  No_common_elements(1,2)==0 &&
No_common_elements(1,1)==1
                holdsamples(testspectra,permutations)=1;
            else
                holdsamples(testspectra,permutations)=0;
            end
            if permutations==6 && testspectra==16
                holdpermutations{1,runs}=holdsamples;
            else
            end
        end
    end
end
% Results are saved
save('results_1_vs1_65nm_pos2_comp_2.mat','holdpermutations')
```

C.8 Convolutional neural networks multiclass classification

The code in this section carries out multiclass classification with Convolutional Neural Networks (CNN). Training and testing spectra are transformed from signals into images using the short time Fourier transform (STFT). Those images are resized to 50 by 50 pixel images. The training images train the network. Stochastic gradient descent with momentum (SGDM) and Bayesian optimisation is applied. The hyperparameters optimised are “InitialLearnRate” the initial learning rate, “Momentum” which is the momentum applied in SGDM, and “L2Regularization” which is a parameter used in the black box loss function to reduce overfitting. Once the network is trained testing is conducted, the predicted classes are compared against the known true classes to gauge correctness. Finally, the results are saved.

Code:

```
clear all
close all
%% Preallocation of variables
holdresults=zeros(1,6);
holdresults_frac=zeros(1,6);
overholdresults{1,5}=[];
overholdresults_frac{1,5}=[];
%% Initialize spectrogram variables
window=2;
noverlap=[];
%% Seed random number generator
rng(1)
%% Load in data
load('Drug_input_spectra_65nm_pos2.mat')
%% Load indexes for testing
load('drug_indexes_65nm_pos2.mat')
%% Initialise indexes
drug1_train=train1;
drug1_test=test1;
drug2_train=train2;
```

```
drug2_test=test2;
drug3_train=train3;
drug3_test=test3;
drug4_train=train4;
drug4_test=test4;
for runs=1:5
   close all
   class_1_training_index=drug1_train(runs,:);
   class_1_testing_index=drug1_test(runs,:);
   class_2_training_index=drug2_train(runs,:);
   class_2_testing_index=drug2_test(runs,:);
   class_3_training_index=drug3_train(runs,:);
   class_3_testing_index=drug3_test(runs,:);
   class_4_training_index=drug4_train(runs,:);
   class_4_testing_index=drug4_test(runs,:);
   trainingsize=size(class_1_training_index,2);
   testingsize=size(class_1_testing_index,2);
   %% Generate training images
   class1=a(class_1_training_index,:);
   for index_index=1:trainingsize
      input=class1(index_index,:);
      c=spectrogram(input,window,noverlap);
      c=abs(c);
      %% Normalisation
      holdmax=max(c);
      holdmax2=max(holdmax);
      holdmax3=c./holdmax2;
      c=holdmax3;
      %% Resize to a 50x50 image
      holdi=imresize(c,[50 50 ]);
      holdj=rescale(holdi,0,1);
      name1='950spectra1';
      name2=num2str(index_index);
      name3='.jpg';
      name4=strcat(name1,name2,name3);
      fullFileName =
fullfile('C:\Users\dex40\Documents\MATLAB\2019_01_29\data_1vsall\training_folder
\class1', name4);
      imwrite(uint8(256*holdj),parula(256),fullFileName)
   end
   class2=a(class_2_training_index,:);
   for index_index=1:trainingsize
      input=class2(index_index,:);
      c=spectrogram(input,window,noverlap);
      c=abs(c);
      %% Normalisation
      holdmax=max(c);
      holdmax2=max(holdmax);
      holdmax3=c./holdmax2;
      c=holdmax3;
```

```
        %% Resize to a 50x50 image
        holdi=imresize(c,[50 50 ]);
        holdj=rescale(holdi,0,1);
        name1='950spectra1';
        name2=num2str(index_index);
        name3='.jpg';
        name4=strcat(name1,name2,name3);
        fullFileName =
fullfile('C:\Users\dex40\Documents\MATLAB\2019_01_29\data_1vsall\training_folder
\class2', name4);
        imwrite(uint8(256*holdj),parula(256),fullFileName)
    end
    class3=a(class_3_training_index,:);
    for index_index=1:trainingsize
        input=class3(index_index,:);
        c=spectrogram(input,window,noverlap);
        c=abs(c);
        %% Normalisation
        holdmax=max(c);
        holdmax2=max(holdmax);
        holdmax3=c./holdmax2;
        c=holdmax3;
        %% Resize to a 50x50 image
        holdi=imresize(c,[50 50 ]);
        holdj=rescale(holdi,0,1);
        name1='950spectra1';
        name2=num2str(index_index);
        name3='.jpg';
        name4=strcat(name1,name2,name3);
        fullFileName =
fullfile('C:\Users\dex40\Documents\MATLAB\2019_01_29\data_1vsall\training_folder
\class3', name4);
        imwrite(uint8(256*holdj),parula(256),fullFileName)

    end
    %%
    class4=a(class_4_training_index,:);
    for index_index=1:trainingsize
        input=class4(index_index,:);
        c=spectrogram(input,window,noverlap);
        c=abs(c);
        %% Normalisation
        holdmax=max(c);
        holdmax2=max(holdmax);
        holdmax3=c./holdmax2;
        c=holdmax3;
        %% Resize to a 50x50 image
        holdi=imresize(c,[50 50 ]);
        holdj=rescale(holdi,0,1);
        name1='950spectra1';
```

```
        name2=num2str(index_index);
        name3='.jpg';
        name4=strcat(name1,name2,name3);
        fullFileName =
fullfile('C:\Users\dex40\Documents\MATLAB\2019_01_29\data_1vsall\training_folder
\class4', name4);
        imwrite(uint8(256*holdj),parula(256),fullFileName)
    end
    %% Generate testing images
    grnobj = a(class_1_testing_index,:);
    for index_index=1:testingsize
        input=grnobj(index_index,:);
        c=spectrogram(input,window,noverlap);
        c=abs(c);
        %% Normalisation
        holdmax=max(c);
        holdmax2=max(holdmax);
        holdmax3=c./holdmax2;
        c=holdmax3;
        %% Resize to a 50x50 image
        holdi=imresize(c,[50 50 ]);
        holdj=rescale(holdi,0,1);
        name1='950spectra1';
        name2=num2str(index_index);
        name3='.jpg';
        name4=strcat(name1,name2,name3);
        fullFileName =
fullfile('C:\Users\dex40\Documents\MATLAB\2019_01_29\data_1vsall\testing_folder\
class1', name4);
        imwrite(uint8(256*holdj),parula(256),fullFileName)
    end
    redobj = a(class_2_testing_index,:);
    for index_index=1:testingsize
        input=redobj(index_index,:);
        c=spectrogram(input,window,noverlap);
        c=abs(c);
        %% Normalisation
        holdmax=max(c);
        holdmax2=max(holdmax);
        holdmax3=c./holdmax2;
        c=holdmax3;
        %% Resize to a 50x50 image
        holdi=imresize(c,[50 50 ]);
        holdj=rescale(holdi,0,1);
        name1='950spectra1';
        name2=num2str(index_index);
        name3='.jpg';
        name4=strcat(name1,name2,name3);
```

```
        fullFileName =
fullfile('C:\Users\dex40\Documents\MATLAB\2019_01_29\data_1vsall\testing_folder\
class2', name4);
        imwrite(uint8(256*holdj),parula(256),fullFileName)
    end
    bluob=a(class_3_testing_index,:);
    for index_index=1:testingsize
        input=bluob(index_index,:);
        c=spectrogram(input,window,noverlap);
        c=abs(c);
        %% Normalisation
        holdmax=max(c);
        holdmax2=max(holdmax);
        holdmax3=c./holdmax2;
        c=holdmax3;
        %% Resize to a 50x50 image
        holdi=imresize(c,[50 50 ]);
        holdj=rescale(holdi,0,1);
        name1='950spectra1';
        name2=num2str(index_index);
        name3='.jpg';
        name4=strcat(name1,name2,name3);
        fullFileName =
fullfile('C:\Users\dex40\Documents\MATLAB\2019_01_29\data_1vsall\testing_folder\
class3', name4);
        imwrite(uint8(256*holdj),parula(256),fullFileName)
    end
    yelobj=a(class_4_testing_index,:);
    for index_index=1:testingsize
        input=yelobj(index_index,:);
        c=spectrogram(input,window,noverlap);
        c=abs(c);
        %% Normalisation
        holdmax=max(c);
        holdmax2=max(holdmax);
        holdmax3=c./holdmax2;
        c=holdmax3;
        %% Resize to a 50x50 image
        holdi=imresize(c,[50 50 ]);
        holdj=rescale(holdi,0,1);
        name1='950spectra1';
        name2=num2str(index_index);
        name3='.jpg';
        name4=strcat(name1,name2,name3);
        fullFileName =
fullfile('C:\Users\dex40\Documents\MATLAB\2019_01_29\data_1vsall\testing_folder\
class4', name4);
        imwrite(uint8(256*holdj),parula(256),fullFileName)
    end
    %% Here datastores are initialised
```

```
    drugTest = imageDatastore('C:\Users\dex40\Documents\MATLAB\2019_01_29\data_1vsall\testing_folder','IncludeSubfolders',true,'LabelSource','foldernames');
    predrugTrain = imageDatastore('C:\Users\dex40\Documents\MATLAB\2019_01_29\data_1vsall\training_folder','IncludeSubfolders',true,'LabelSource','foldernames');
    [drugTrain,drugValidation] = splitEachLabel(predrugTrain,12)
    numClasses = numel(categories(predrugTrain.Labels));
    %% Here the network is initialised
    layers = [
        imageInputLayer([50 50 3],"Name","imageinput")
        convolution2dLayer([5 5],20,"Name","conv")
        reluLayer("Name","relu")
        maxPooling2dLayer([2 2],"Name","maxpool","Stride",[2 2])
        fullyConnectedLayer(10,"Name","fc")
        softmaxLayer("Name","softmax")
        classificationLayer("Name","classoutput")];
    lgraph = layerGraph(layers);
    lgraph = removeLayers(lgraph, {'fc','softmax','classoutput'});
    newLayers = [
        fullyConnectedLayer(numClasses,'Name','fc','WeightLearnRateFactor',1,'BiasLearnRateFactor',1)
        softmaxLayer('Name','softmax')
        classificationLayer('Name','classoutput')];
    lgraph = addLayers(lgraph,newLayers);
    lgraph = connectLayers(lgraph,'maxpool','fc');
    layers = lgraph.Layers;
    connections = lgraph.Connections;
    lgraph = createLgraphUsingConnections(layers,connections);
    optimVars = [
        optimizableVariable('InitialLearnRate',[1e-4 1e0],'Transform','log')
        optimizableVariable('Momentum',[0 1])
        optimizableVariable('L2Regularization',[1e-10 1e-2],'Transform','log')];
    XTrain=drugTrain;
    YTrain=drugTrain.Labels;
    XValidation=drugValidation;
    YValidation=drugValidation.Labels;
    XTest=drugTest;
    YTest=drugTest.Labels;
    ObjFcn = makeObjFcn(XTrain,YTrain,XValidation,YValidation,lgraph);
    %% Here Bayesian optimization [1-5] is carried out
    BayesObject = bayesopt(ObjFcn,optimVars,...
        'MaxObj',100,...
        'MaxTime',24*60*60,...
        'IsObjectiveDeterministic',false,...
        'AcquisitionFunctionName','expected-improvement-plus',...
        'UseParallel',false,'Verbose', 0, 'PlotFcn', []);
    bestIdx = BayesObject.IndexOfMinimumTrace(end);
    fileName = BayesObject.UserDataTrace{bestIdx};
```

```
    savedStruct = load(fileName);
    valError = savedStruct.valError
    [YPredicted,probs] = classify(savedStruct.trainedNet,XTest);
    testError = 1 - mean(YPredicted == YTest)
    NTest = numel(YTest);
    testErrorSE = sqrt(testError*(1-testError)/NTest);
    testError95CI = [testError - 1.96*testErrorSE, testError + 1.96*testErrorSE]
    figure
    [cmat,classNames] = confusionmat(YTest,YPredicted);
    h = heatmap(classNames,classNames,cmat);
    xlabel('Predicted Class');
    ylabel('True Class');
    title('Confusion Matrix');
    %% Here the number of correctly classified drugs is calculated
    numCorrect = nnz(YPredicted == YTest);
    fracCorrect = numCorrect/numel(YPredicted);
    clear class_1_training_index
    clear class_2_training_index
    clear class_1_testing_index
    clear class_2_testing_index
    clear class_3_training_index
    clear class_4_training_index
    clear class_3_testing_index
    clear class_4_testing_index
    overholdresults{1,runs}=numCorrect;
    overholdresults_frac{1,runs}=fracCorrect;
end
% Results are saved
save('baysian_optimised_multiclass_CNN_drugs_65nm_pos2_results7layer_cnn.mat','o
verholdresults','overholdresults_frac')
function ObjFcn = makeObjFcn(XTrain,YTrain,XValidation,YValidation,lgraph)
ObjFcn = @valErrorFun;
    function [valError,cons,fileName] = valErrorFun(optVars)
        numClasses = numel(unique(YTrain));
        layers = lgraph ;
        miniBatchSize = 1200;
        validationFrequency = 10;
        options = trainingOptions('sgdm',...
            'InitialLearnRate',optVars.InitialLearnRate,...
            'Momentum',optVars.Momentum,...
            'MaxEpochs',10, ...
            'L2Regularization',optVars.L2Regularization,...
            'Verbose',false,...
            'ValidationData',{XValidation,YValidation},...
            'ValidationPatience',Inf,...
            'ValidationFrequency',validationFrequency);
        trainedNet = trainNetwork(XTrain,layers,options);
        close(findall(groot,'Tag','NNET_CNN_TRAININGPLOT_FIGURE'))
        YPredicted = classify(trainedNet,XValidation);
        valError = 1 - mean(YPredicted == YValidation);
```

```
        fileName =convertCharsToStrings(pwd)+"\"+num2str(valError) + ".mat";
        save(fileName,'trainedNet','valError','options')
        cons = [];
    end
end
%% References
%[1]MathWorks.Support Vector Machines for Binary Classification.2017[Cited
13/02/1018]. Avialable from:https://www.mathworks.com/help/stats/support-vector-
machines-for-binary-classification.html#bsr5o1q
%[2]Hastie T, Tibshirani R, Friedman J. The elements of statistical learning: data
mining, inference, and prediction. Springer Science & Business Media; 2009 Aug 26.
%[3]Christianini, N, and J Shawe-Taylor. An Introduction to Support Vector Machines
and Other Kernel-Based Learning Methods. Cambridge, UK: Cambridge University
Press, 2000.
%[4]Fan RE, Chen PH, Lin CJ. Working set selection using second order information
for training support vector machines. Journal of machine learning research.
2005;6(Dec):1889-918.
%[5]Kecman V, Huang TM, Vogt M. Iterative single data algorithm for training kernel
machines from huge data sets: Theory and performance. InSupport vector machines:
Theory and Applications 2005 Jun 21 (pp. 255-274). Springer, Berlin, Heidelberg.
```

C.9 Convolutional neural networks 1 vs 1 classification

The code in this section carries out 1 vs 1classification with CNNs. Training and testing spectra are transformed from signals into images using the STFT. Those images are resized to 50 by 50 pixel images. The training images train the network. SGDM and Bayesian optimisation is applied. The hyperparameters optimised are “InitialLearnRate” the initial learning rate, “Momentum” which is the momentum applied in SGDM, and “L2Regularization” which is a parameter used in the black box loss function to reduce overfitting. Once the network is trained testing is conducted, the predicted classes are compared against the known true classes to gauge correctness. Finally, the results are saved.

Code:

```
clear all
close all
%% Preallocation of variables
holdresults=zeros(1,6);
holdresults_frac=zeros(1,6);
overholdresults{1,5}=[];
overholdresults_frac{1,5}=[];
%% Initialize spectrogram variables
window=2;
noverlap=[];
%% Seed random number generator
rng(1)
%% Load in data
load('Drug_input_spectra_65nm_pos2.mat')
%% Load indexes for testing
load('drug_indexes_65nm_pos2.mat')
%% load indexes for testing
rng(1)
%% Initialise indexes
drug1_train=train1;
drug1_test=test1;
drug2_train=train2;
drug2_test=test2;
drug3_train=train3;
drug3_test=test3;
drug4_train=train4;
drug4_test=test4;
for j=1:5
    for i=1:6
        close all
        if i==1
            class_1_training_index=drug1_train(j,:);
            class_1_testing_index=drug1_test(j,:);
            class_2_training_index=drug2_train(j,:);
            class_2_testing_index=drug2_test(j,:);
        elseif i==2
            class_1_training_index=drug1_train(j,:);
            class_1_testing_index=drug1_test(j,:);
            class_2_training_index=drug3_train(j,:);
            class_2_testing_index=drug3_test(j,:);
        elseif i==3
            class_1_training_index=drug1_train(j,:);
            class_1_testing_index=drug1_test(j,:);
            class_2_training_index=drug4_train(j,:);
            class_2_testing_index=drug4_test(j,:);
        elseif i==4
            class_1_training_index=drug2_train(j,:);
            class_1_testing_index=drug2_test(j,:);
            class_2_training_index=drug3_train(j,:);
            class_2_testing_index=drug3_test(j,:);
        elseif i==5
            class_1_training_index=drug2_train(j,:);
```

```
            class_1_testing_index=drug2_test(j,:);
            class_2_training_index=drug4_train(j,:);
            class_2_testing_index=drug4_test(j,:);
        elseif i==6
            class_1_training_index=drug3_train(j,:);
            class_1_testing_index=drug3_test(j,:);
            class_2_training_index=drug4_train(j,:);
            class_2_testing_index=drug4_test(j,:);
        end
        trainingsize=size(class_1_training_index,2);
        testingsize=size(class_1_testing_index,2);
        class1=a(class_1_training_index,:);
        for index_index=1:trainingsize
            input=class1(index_index,:);
            c=spectrogram(input,window,noverlap);
            c=abs(c);
            %% Normalisation
            holdmax=max(c);
            holdmax2=max(holdmax);
            holdmax3=c./holdmax2;
            c=holdmax3;
            %% Resize to a 50x50 image
            holdi=imresize(c,[50 50 ]);
            holdj=rescale(holdi,0,1);
            name1='950spectra1';
            name2=num2str(index_index);
            name3='.jpg';
            name4=strcat(name1,name2,name3);
            fullFileName = fullfile('C:\Users\dex40\Documents\MATLAB\2019_01_24\data_1vs1_bin\training_folder\class1', name4);
            imwrite(uint8(256*holdj),parula(256),fullFileName)
        end
        class2=a(class_2_training_index,:);
        for index_index=1:trainingsize
            input=class2(index_index,:);
            c=spectrogram(input,window,noverlap);
            c=abs(c);
            %% Normalisation
            holdmax=max(c);
            holdmax2=max(holdmax);
            holdmax3=c./holdmax2;
            c=holdmax3;
            %% Resize to a 50x50 image
            holdi=imresize(c,[50 50 ]);
            holdj=rescale(holdi,0,1);
            name1='950spectra1';
            name2=num2str(index_index);
            name3='.jpg';
            name4=strcat(name1,name2,name3);
            fullFileName = fullfile('C:\Users\dex40\Documents\MATLAB\2019_01_24\data_1vs1_bin\training_folder\class2', name4);
            imwrite(uint8(256*holdj),parula(256),fullFileName)
```

```
        end
        %% Load testing data
        grnobj = a(class_1_testing_index,:);
        for index_index=1:testingsize
            input=grnobj(index_index,:);
            c=spectrogram(input,window,noverlap);
            c=abs(c);
            %% Normalisation
            holdmax=max(c);
            holdmax2=max(holdmax);
            holdmax3=c./holdmax2;
            c=holdmax3;
           %% Resize to a 50x50 image
            holdi=imresize(c,[50 50 ]);
            holdj=rescale(holdi,0,1);
            name1='950spectra1';
            name2=num2str(index_index);
            name3='.jpg';
            name4=strcat(name1,name2,name3);
            fullFileName =
fullfile('C:\Users\dex40\Documents\MATLAB\2019_01_24\data_1vs1_bin\testing_folder\class1', name4);
            imwrite(uint8(256*holdj),parula(256),fullFileName)
        end
        redobj = a(class_2_testing_index,:);
        for index_index=1:testingsize
            input=redobj(index_index,:);
            c=spectrogram(input,window,noverlap);
            c=abs(c);
            %% Normalisation
            holdmax=max(c);
            holdmax2=max(holdmax);
            holdmax3=c./holdmax2;
            c=holdmax3;
            %% Resize to a 50x50 image
            holdi=imresize(c,[50 50 ]);
            holdj=rescale(holdi,0,1);
            name1='950spectra1';
            name2=num2str(index_index);
            name3='.jpg';
            name4=strcat(name1,name2,name3);
            fullFileName =
fullfile('C:\Users\dex40\Documents\MATLAB\2019_01_24\data_1vs1_bin\testing_folder\class2', name4);
            imwrite(uint8(256*holdj),parula(256),fullFileName)
        end
        %% Here datastores are initialised
        drugTest =
imageDatastore('C:\Users\dex40\Documents\MATLAB\2019_01_24\data_1vs1_bin\testing_folder','IncludeSubfolders',true,'LabelSource','foldernames');
        predrugTrain =
imageDatastore('C:\Users\dex40\Documents\MATLAB\2019_01_24\data_1vs1_bin\training_folder','IncludeSubfolders',true,'LabelSource','foldernames');
        [drugTrain,drugValidation] = splitEachLabel(predrugTrain,12)
        numClasses = numel(categories(predrugTrain.Labels));
```

```
%% Here the network is initialised
layers = [
   imageInputLayer([50 50 3],"Name","imageinput")
   convolution2dLayer([5 5],20,"Name","conv")
   reluLayer("Name","relu")
   maxPooling2dLayer([2 2],"Name","maxpool","Stride",[2 2])
   fullyConnectedLayer(10,"Name","fc")
   softmaxLayer("Name","softmax")
   classificationLayer("Name","classoutput")];
lgraph = layerGraph(layers);
lgraph = removeLayers(lgraph, {'fc','softmax','classoutput'});
newLayers = [

fullyConnectedLayer(numClasses,'Name','fc','WeightLearnRateFactor',1,'BiasLearnRateFactor',
1)
   softmaxLayer('Name','softmax')
   classificationLayer('Name','classoutput')];
lgraph = addLayers(lgraph,newLayers);
lgraph = connectLayers(lgraph,'maxpool','fc');
layers = lgraph.Layers;
connections = lgraph.Connections;
lgraph = createLgraphUsingConnections(layers,connections);
optimVars = [
   optimizableVariable('InitialLearnRate',[1e-4 1e0],'Transform','log')
   optimizableVariable('Momentum',[0 1])
   optimizableVariable('L2Regularization',[1e-10 1e-2],'Transform','log')];
XTrain=drugTrain;
YTrain=drugTrain.Labels;
XValidation=drugValidation;
YValidation=drugValidation.Labels;
XTest=drugTest;
YTest=drugTest.Labels;
ObjFcn = makeObjFcn(XTrain,YTrain,XValidation,YValidation,lgraph);
%% Here Bayesian optimization [1-5] is carried out
BayesObject = bayesopt(ObjFcn,optimVars,...
   'MaxObj',100,...
   'MaxTime',24*60*60,...
   'IsObjectiveDeterministic',false,...
   'AcquisitionFunctionName','expected-improvement-plus',...
   'UseParallel',false,'Verbose', 0, 'PlotFcn', []);
bestIdx = BayesObject.IndexOfMinimumTrace(end);
fileName = BayesObject.UserDataTrace{bestIdx};
savedStruct = load(fileName);
valError = savedStruct.valError
[YPredicted,probs] = classify(savedStruct.trainedNet,XTest);
testError = 1 - mean(YPredicted == YTest)
NTest = numel(YTest);
testErrorSE = sqrt(testError*(1-testError)/NTest);
testError95CI = [testError - 1.96*testErrorSE, testError + 1.96*testErrorSE]
figure
[cmat,classNames] = confusionmat(YTest,YPredicted);
h = heatmap(classNames,classNames,cmat);
xlabel('Predicted Class');
ylabel('True Class');
title('Confusion Matrix');
```

```
        %% Here the number of correctly classified drugs is calculated
        numCorrect = nnz(YPredicted == YTest);
        fracCorrect = numCorrect/numel(YPredicted);
        holdresults(:,i)=numCorrect;
        holdresults_frac(:,i)=fracCorrect;
        clear class_1_training_index
        clear class_2_training_index
        clear class_1_testing_index
        clear class_2_testing_index
        if i==6
            overholdresults{1,j}=holdresults;
            overholdresults_frac{1,j}=holdresults_frac;
        else
        end
    end
end
% Results are saved
save('template_baysian_optimised_1vs1_CNN_65nm_pos2_results_7layer_cnn.mat','overholdresults','overholdresults_frac')
function ObjFcn = makeObjFcn(XTrain,YTrain,XValidation,YValidation,lgraph)
ObjFcn = @valErrorFun;
    function [valError,cons,fileName] = valErrorFun(optVars)
        numClasses = numel(unique(YTrain));
        layers = lgraph ;
        miniBatchSize = 1200;
        validationFrequency = 10;
        options = trainingOptions('sgdm',...
            'InitialLearnRate',optVars.InitialLearnRate,...
            'Momentum',optVars.Momentum,...
            'MaxEpochs',10, ...
            'L2Regularization',optVars.L2Regularization,...
            'Verbose',false,...
            'ValidationData',{XValidation,YValidation},...
            'ValidationPatience',Inf,...
            'ValidationFrequency',validationFrequency);
        trainedNet = trainNetwork(XTrain,layers,options);
        close(findall(groot,'Tag','NNET_CNN_TRAININGPLOT_FIGURE'))
        YPredicted = classify(trainedNet,XValidation);
        valError = 1 - mean(YPredicted == YValidation);
        fileName =convertCharsToStrings(pwd)+"\"+num2str(valError) + ".mat";
        save(fileName,'trainedNet','valError','options')
        cons = [];
    end
end
%% References
%[1] MathWorks.Support Vector Machines for Binary Classification.2017[Cited 13/02/1018]. Avialable from:https://www.mathworks.com/help/stats/support-vector-machines-for-binary-classification.html#bsr5o1q
%[2] Hastie T, Tibshirani R, Friedman J. The elements of statistical learning: data mining, inference, and prediction. Springer Science & Business Media; 2009 Aug 26.
%[3] Christianini, N, and J Shawe-Taylor. An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods. Cambridge, UK: Cambridge University Press, 2000.
%[4] Fan RE, Chen PH, Lin CJ. Working set selection using second order information for training support vector machines. Journal of machine learning research. 2005;6(Dec):1889-918.
```

%[5] Kecman V, Huang TM, Vogt M. Iterative single data algorithm for training kernel machines from huge data sets: Theory and performance. InSupport vector machines: Theory and Applications 2005 Jun 21 (pp. 255-274). Springer, Berlin, Heidelberg.

D. Identification of emission lines observed in the recorded experimental spectra.

A large amount of spectroscopic data was gathered during the experimental investigations contributing to this thesis. This was glossed over somewhat as the aim of the studies were to classify the different samples, and to assess the accuracy of the techniques used to do this. As such the patterns present in the spectra were of greater interest than the individual emission lines. However, the emission lines that were encountered during these studies were identified using the National Institute of Standards and Technology's atomic spectra database [1]. These originated mainly from lowly charged O, C and N. Tables D.1 to D.5 identity the strongest spectral emission lines observed in chapter 4, while those from chapter 5 are listed in tables D.6- D.8. Finally, table D.9 contains the lines present in the target and lens position optimisation procedure undertaken in chapter 2.

Table D.1: Emission lines identified in the 105 nm region spectra from chapter 4.

Ion	Wavelength (nm)	Lower Level Conf., Term, J	Upper Level Conf., Term, J
C III	97.703	$2s^2$, ^1S, 0	2s2p, 1P°, 1
C II	101.037	$2s2p^2$, ^4P, 5/2	$2p^3$, $^4S°$, 3/2
O I	102.5762	$2s^22p^4$, 3P, 2	$2s^22p^3(^4S°)3d$, $^3D°$, 3
O I	102.5762	$2s^22p^4$, ^3P, 2	$2s^22p^3(^4S°)3d$, $^3D°$, 1
O I	102.5762	$2s^22p^4$, ^3P, 2	$2s^22p^3(^4S°)3d$, $^3D°$, 2
C I	111.5393	$2s^22p^2$, ^3P, 2	$2s^22p(^{2P°}{}_{1/2})10d$, 2[5/2]°, 2
O VI	103.1912	$1s^22s$, ^2S, 1/2	$1s^22p$, $^2P°$, 3/2
O VI	103.7613	$1s^22s$, ^2S, 1/2	$1s^22p$, $^2P°$, 1/2

Table D.2: Emission lines identified in the 85 nm region spectra from chapter 4.

Ion	Wavelength (nm)	Lower Level Conf., Term, J	Upper Level Conf., Term, J
O IV	79.0109	$2s^{2}2p$, $^{2}P^{\circ}$, 3/2	$2s2p^{2}$, ^{2}D, 3/2
O III	79.0115	$2s^{2}2p(^{2}P^{\circ})3s$, $^{3}P^{\circ}$, 2	$2s2p^{2}(^{2}D)3s$, ^{3}D, 3
O IV	79.0199	$2s^{2}2p$, $^{2}P^{\circ}$, 3/2	$2s2p^{2}$, ^{2}D, 5/2
O V	77.4518	$1s^{2}2s2p$, $^{1}P^{\circ}$, 1	$1s^{2}2p^{2}$, ^{1}S, 0
N III	77.4357	$2s^{2}3p$, $^{2}P^{\circ}$, 1/2	$2s2p(3P^{\circ})4p$, ^{2}P, 3/2
N III	77.4572	$2s^{2}3p$, $^{2}P^{\circ}$, 3/2	$2s2p(^{3}P^{\circ})4p$, ^{2}P, 3/2
O III	83.5292	$2s^{2}2p^{2}$, ^{3}P, 2	$2s2p^{3}$, $^{3}D^{\circ}$, 3
N III	83.5193	$2s2p(^{3}P^{\circ})3p$, ^{4}D, 5/2	$2s2p(^{3}P^{\circ})7d$, $^{4}P^{\circ}$, 3/2
N III	83.5193	$2s2p(^{3}P^{\circ})3p$, ^{4}D, 5/2	$2s2p(^{3}P^{\circ})7d$, $^{4}P^{\circ}$, 5/2
N III	83.5275	$2s2p(^{3}P^{\circ})3p$, ^{4}D, 3/2	$2s2p(^{3}P^{\circ})7d$, $^{4}D^{\circ}$, 5/2
N III	83.5275	$2s2p(^{3}P^{\circ})3p$, ^{4}D, 3/2	$2s2p(^{3}P^{\circ})7d$, $^{4}D^{\circ}$, 3/2
N III	83.5275	$2s2p(^{3}P^{\circ})3p$, ^{4}D, 3/2	$2s2p(^{3}P^{\circ})7d$, $^{4}D^{\circ}$, 1/2
N II	85.8376	$2s^{2}2p^{2}$, ^{1}S, 0	$2s^{2}2p3s$, $^{1}P^{\circ}$, 1
C II	85.8559	$2s^{2}2p$, $^{2}P^{\circ}$, 3/2	$2s^{2}3s$, ^{2}S, 1/2
C II	90.4142	$2s^{2}2p$, $^{2}P^{\circ}$, 3/2	$2s2p^{2}$, ^{2}P, 3/2
N IV	92.3220	$1s^{2}2s2p$, $^{3}P^{\circ}$, 2	$1s^{2}2p^{2}$, ^{3}P, 2
N IV	92.3220	$1s^{2}2s2p$, $^{3}P^{\circ}$, 2	$1s^{2}2p^{2}$, ^{3}P, 2
O IV	92.3367	$2s2p^{2}$, ^{2}P, 3/2	$2p^{3}$, $^{2}P^{\circ}$, 3/2

Table D.3: Emission lines identified in the 75 nm region spectra from chapter 4.

Ion	Wavelength (nm)	Lower Level Conf., Term, J	Upper Level Conf., Term, J
O III	70.2822	$2s^22p^2$, ^3P, 1	$2s2p^3$, $^3P^\circ$, 0
O III	70.2899	$2s^22p^2$, ^3P, 1	$2s2p^3$, $^3P^\circ$, 1
O III	70.2899	$2s^22p^2$, ^3P, 1	$2s2p^3$, $^3P^\circ$, 2
N V	76.0413	$1s^24p$, $^2P^\circ$, 1/2	$1s^28d$, ^2D, 3/2
N V	76.0596	$1s^24p$, $^2P^\circ$, 3/2	$1s^28d$, ^2D, 5/2
N V	76.0603	$1s^24p$, $^2P^\circ$, 3/2	$1s^28d$, ^2D, 3/2
O V	76.0445	$1s^22s2p$, $^3P^\circ$, 2	$1s^22p2$, ^3P, 2
O V	76.0446	$1s^22s2p$, $^3P^\circ$, 2	$1s^22p2$, ^3P, 2
N III	77.4572	$2s^23p$, $^2P^\circ$, 3/2	$2s2p(^3P^\circ)4p$, ^2P, 3/2
O V	77.4518	$1s^22s2p$, $^1P^\circ$, 1	$1s^22p^2$, ^1S, 0
O III	79.0556	$2s^22p(^2P^\circ)3s$,$^3P^\circ$, 2	$2s2p^2(^2D)3s$, ^3D, 2
C II	80.6830	$2s2p^2$, ^4P, 3/2	$2s2p(^3P^\circ)3s$, $^4P^\circ$, 1/2
C II	80.6861	$2s2p^2$, ^4P, 5/2	$2s2p(^3P^\circ)3s$, $^4P^\circ$, 3/2

Table D.4: Emission lines identified in the 55 nm region spectra from chapter 4.

Ion	Wavelength (nm)	Lower Level Conf., Term, J	Upper Level Conf., Term, J
O VII	53.8851	1s4p, ^{1}P°, 1	1s6d, ^{1}D, 2
O III	55.5026	2s2p^3, ^{3}D°, 1	2s^{2}2p(^{2}P°)3p, ^{3}P, 0
O III	57.4786	2s2p^3, ^{3}D°, 3	2s^{2}2p(^{2}P°)3p, ^{3}D, 2
O III	57.4880	2s2p^3, ^{3}D°, 2	2s^{2}2p(^{2}P°)3p, ^{3}D, 2
O III	57.4895	2s2p^3, ^{3}D°, 1	2s^{2}2p(^{2}P°)3p, ^{3}D, 2
O III	61.004	2s2p^3, ^{3}D°, 2	2p^4, ^{3}P, 1
O III	61.0057	2s2p^3, ^{3}D°, 1	2p^4, ^{3}P, 1
O IV	61.6952	2s2p^2, ^{2}D, 5/2	2p^3, ^{2}P°, 3/2
O IV	61.7005	2s2p^2, ^{2}D, 3/2	2p^3, ^{2}P°, 3/2
O IV	61.7036	2s2p^2, ^{2}D, 3/2	2p^3, ^{2}P°, 1/2
C VI	52.02773	3p, ^{2}P°, 1/2	4d, ^{2}D, 3/2
C VI	52.02981	3s, ^{2}S, 1/2	4p, ^{2}P°, 3/2
O II	52.0173	2s^{2}2p^3, ^{2}P°, 3/2	2s^{2}2p^2(^{3}P)3d, ^{4}D, 1/2
O II	52.0179	2s^{2}2p^3, ^{2}P°, 1/2	2s^{2}2p^2(^{3}P)3d, ^{4}D, 1/2

Table D.5: Emission lines identified in the 45 nm region spectra from chapter 4.

Ion	Wavelength (nm)	Lower Level Conf., Term, J	Upper Level Conf., Term, J
C V	37.4211	1s3p, ^{3}P°, 1	1s8s, ^{3}S, 1
C V	37.4213	1s3p, ^{3}P°, 0	1s8s, ^{3}S, 1
C V	37.4267	1s3p, ^{3}P°, 2	1s8s, ^{3}S, 1
N III	37.4204	2s^{2}2p, ^{2}P°, 1/2	2s^{2}3d, ^{2}D, 3/2
O III	37.433	2s^{2}2p2, ^{3}P, 1	2s^{2}2p(^{2}P°)3s, ^{3}P°, 0
C IV	38.403	1s^{2}2p, ^{2}P°, 1/2	1s^{2}3d, ^{2}D, 3/2
C IV	38.418	1s^{2}2p, ^{2}P°, 3/2	1s^{2}3d, ^{2}D, 5/2
C IV	38.4190	1s^{2}2p, ^{2}P°, 3/2	1s^{2}3d, ^{2}D, 3/2
C IV	41.971	1s^{2}2p, ^{2}P°, 3/2	1s^{2}3s, ^{2}S, 1/2
C III	45.963	1s^{2}2s2p, ^{3}P°, 2	1s^{2}2s3d, ^{3}D, 3
C III	45.9633	1s^{2}2s2p, ^{3}P°, 2	1s^{2}2s3d, ^{3}D, 2
C III	45.9635	1s^{2}2s2p, ^{3}P°, 2	1s^{2}2s3d, ^{3}D, 1
C III	49.9583	1s^{2}2p2, ^{3}P, 2	1s^{2}2p(^{2}P°)3d, ^{3}D°, 2
C III	49.9615	1s^{2}2p2, ^{3}P, 2	1s^{2}2p(^{2}P°)3d, ^{3}D°, 1
O VI	51.9723	1s^{2}3d, ^{2}D, 5/2	1s^{2}4f, ^{2}F°, 7/2
O VI	51.9748	1s^{2}3d, ^{2}D, 5/2	1s^{2}4f, ^{2}F°, 5/2

Table D.6: Emission lines identified in the 105 nm region spectra from chapter 5.

Ion	Wavelength (nm)	Lower Level Conf., Term, J	Upper Level Conf., Term, J
N III	97.6905	2s2p(^{3}P°)3p, ^{4}P, 3/2	2p^2(^{3}P)3p, ^{4}D°, 1/2
C III	97.703	1s^{2}2s^2, ^{1}S, 0	1s^{2}2s2p, ^{1}P°, 1
C II	103.7018	2s^{2}2p, ^{2}P°, 3/2	2s^{2}p2, ^{2}S, 1/2
O IV	103.709	2s2p(^{3}P°)3d, ^{4}D°, 7/2	2s2p(^{3}P°)4f, ^{4}D, 7/2
C I	110.840	2s^{2}2p^2, ^{3}P, 2	2s^{2}2p($^{2P^\circ}{}_{1/2}$)14d,^{2}[3/2]°, 1
C I	110.8440	2s^{2}2p^2, ^{3}P, 1	2s^{2}2p($^{2P^\circ}{}_{3/2}$)13d, 2[3/2]°, 1

Table D.7: Emission lines identified in the 65 nm region spectra from chapter 5.

Ion	Wavelength (nm)	Lower Level Conf., Term, J	Upper Level Conf., Term, J
O IV	57.4405	2s^{2}3d, ^{2}D, 3/2	2s2p(^{3}P°)4d, ^{2}D°, 3/2
O IV	57.4460	2s^{2}3d, ^{2}D, 5/2	2s2p(^{3}P°)4d, ^{2}D°, 3/2
O IV	61.6488	2s^{2}3d, ^{2}D, 3/2	2s2p(^{1}P°)3d, ^{2}P°, 3/2
O IV	61.6552	2s^{2}3d, ^{2}D, 5/2	2s2p(^{1}P°)3d, ^{2}P°, 3/2
O IV	61.6572	2s^{2}3d, ^{2}D, 3/2	2s2p(^{1}P°)3d, ^{2}P°, 1/2
O IV	62.4825	2s2p(^{3}P°)3p, ^{4}S, 3/2	2s2p(^{3}P°)5d, ^{4}P°, 5/2
N III	68.5673	2s2p^2, ^{4}P, 3/2	2p^3, ^{2}D°, 3/2
N III	68.5740	2s2p^2, ^{4}P, 3/2	2p^3, ^{2}D°, 5/2
N III	68.5816	2s^{2}2p, ^{2}P°, 3/2	2s2p^2, ^{2}P, 3/2

Table D.8: Emission lines identified in the 65 nm region spectra from chapter 5.

Ion	Wavelength (nm)	Lower Level Conf., Term, J	Upper Level Conf., Term, J
C III	57.4281	$1s^22s2p$, $^1P^\circ$, 1	$1s^22s3d$, ^1D, 2
O IV	57.4138	$2s^23d$, ^2D, 3/2	$2s2p(^3P^\circ)4d$, $^2D^\circ$, 5/2
O III	57.4153	$2s2p^3$, $^3D^\circ$, 2	$2s^22p(^2P^\circ)3p$, ^3D, 3
O IV	57.4193	$2s^23d$, ^2D, 5/2	$2s2p(^3P^\circ)4d$, $^2D^\circ$, 5/2
O IV	62.4617	$2s2p^2$, ^4P, 1/2	$2p^3$, $^4S^\circ$, 3/2
N III	68.5393	$2s2p^2$, ^4P, 1/2	$2p^3$, $^2D^\circ$, 3/2
N III	68.5513	$2s^22p$, $^2P^\circ$, 1/2	$2s2p^2$, ^2P, 1/2
C IV	68.5405	$1s^23d$, ^2D, 3/2	$1s^26p$, $^2P^\circ$, 3/2
C IV	68.5424	$1s^23d$, ^2D, 3/2	$1s^26p$, $^2P^\circ$, 1/2
C IV	68.5455	$1s^23d$, ^2D, 5/2	$1s^26p$, $^2P^\circ$, 3/2
C III	68.5513	$1s^22s^2$, ^1S, 0	$1s^22p^2$, ^1D, 2

Table D.9: Emission lines identified in the 65 nm region spectra from chapter 2.

Ion	Wavelength (nm)	Lower Level Conf., Term, J	Upper Level Conf., Term, J
O V	62.7223	$1s^2 2s3p$, $^3P^\circ$, 0	$1s^2 2s4d$, ^3D, 1
C II	68.7352	$2s^2 2p$, $^2P^\circ$, 3/2	$2s^2 3d$, ^2D, 3/2
C IV	66.0850	$1s^2 3p$, $^2P^\circ$, 1/2	$1s^2 6d$, ^2D, 3/2
C IV	66.0983	$1s^2 3p$, $^2P^\circ$, 3/2	$1s^2 6d$, ^2D, 5/2
C IV	66.0988	$1s^2 3p$, $^2P^\circ$, 3/2	$1s^2 6d$, ^2D, 3/2
O III	65.9536	$2s2p^3$, $^3P^\circ$, 2	$2s^2 2p(^2P^\circ)3p$, ^3D, 2
O III	65.9539	$2s2p^3$, $^3P^\circ$, 1	$2s^2 2p(^2P^\circ)3p$, ^3D, 2
O III	66.0129	$2s2p^3$, $^3P^\circ$, 2	$2s^2 2p(^2P^\circ)3p$, ^3D, 1
O III	66.0133	$2s2p^3$, $^3P^\circ$, 1	$2s^2 2p(^2P^\circ)3p$, ^3D, 1
O III	66.0184	$2s2p^3$, $^3P^\circ$, 0	$2s^2 2p(^2P^\circ)3p$, ^3D, 1
N II	66.0286	$2s^2 2p^2$, ^1D, 2	$2s2p^3$, $^1P^\circ$, 1
N III	66.0551	$2s2p^2$, ^2P, 1/2	$2s2p(^3P^\circ)3s$, $^2P^\circ$, 3/2
N III	66.1032	$2s2p^2$, ^2P, 3/2	$2s2p(^3P^\circ)3s$, $^2P^\circ$, 3/2
N III	66.1046	$2s2p^2$, ^2P, 1/2	$2s2p(^3P^\circ)3s$, $^2P^\circ$, 1/2
O III	65.9536	$2s2p^3$, $^3P^\circ$, 2	$2s^2 2p(^2P^\circ)3p$, ^3D, 2
O III	65.9539	$2s2p^3$, $^3P^\circ$, 1	$2s^2 2p(^2P^\circ)3p$, ^3D, 2
O III	66.0129	$2s2p^3$, $^3P^\circ$, 2	$2s^2 2p(^2P^\circ)3p$, ^3D, 1
O III	66.0133	$2s2p^3$, $^3P^\circ$, 1	$2s^2 2p(^2P^\circ)3p$, ^3D, 1
O III	66.0184	$2s2p^3$, $^3P^\circ$, 0	$2s^2 2p(^2P^\circ)3p$, ^3D, 1
N II	66.0286	$2s^2 2p^2$, ^1D, 2	$2s2p^3$, $^1P^\circ$, 1
C II	68.6488	$2s^2 p^2$, ^2D, 3/2	$2s2p(^3P^\circ)4d$, $^2D^\circ$, 3/2

O IV	68.6687	2s2p(^{3}P°)3d, ^{2}P°, 1/2	2p^2(^{1}D)3d, ^{2}P, 1/2
O IV	68.6687	2s2p(^{3}P°)3d, ^{2}P°, 1/2	2p^2(^{1}D)3d, ^{2}P, 3/2

Reference:

[1] Ralchenko Y, Kramida AE, Reader J. NIST atomic spectra database. National Institute of Standards and Technology, Gaithersburg, MD. Updated Oct 2019. Available from: https://dx.doi.org/10.18434/T4W30F

E. Determination of target surface to optical axis distance for time integrated spatially resolved LIBS.

The position of the target surface with respect to the optical axis for the initial study in chapter 4 was determined from analysis using pressboard targets. Figures E.1 to E.6 show spectra gathered at three sites on the surface of a target where the separation between the target surface and the optic axis of the spectrometer is varied from 0 mm to 2.25 mm. The lens–target distance was kept constant to ensure similar power densities for all spectra. The optimal position selected for the study in chapter 4 was 2 mm as at this distance distinct lines are present, and the continuum was relatively low.

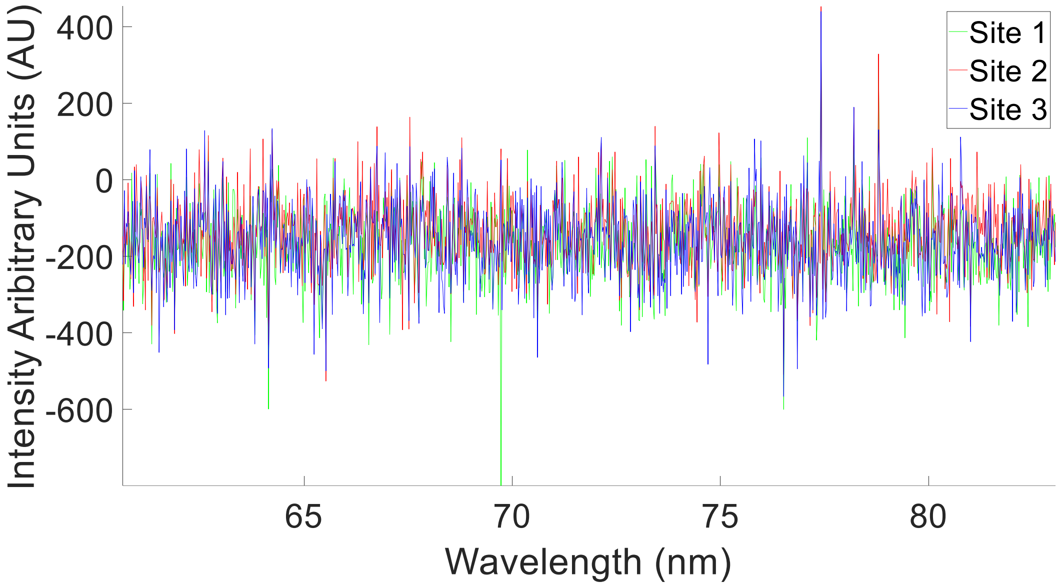


Figure E.1: Single shot spectra taken at three sites on a target surface located 0.00 mm from the optical axis of the spectrometer.

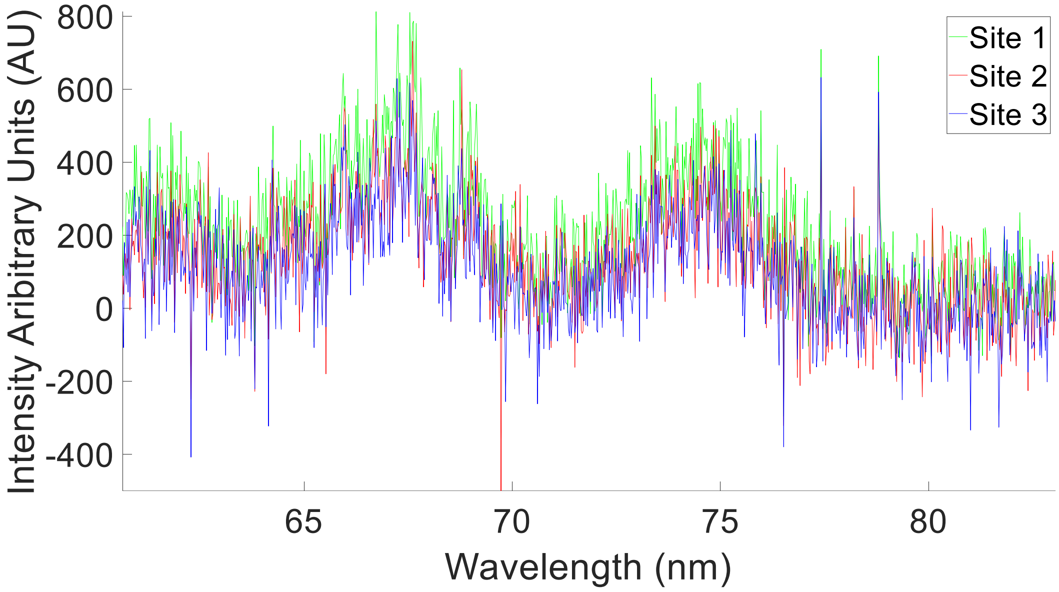


Figure E.2: Single shot spectra taken at three sites on a target surface located 1.00 mm from the optical axis of the spectrometer.

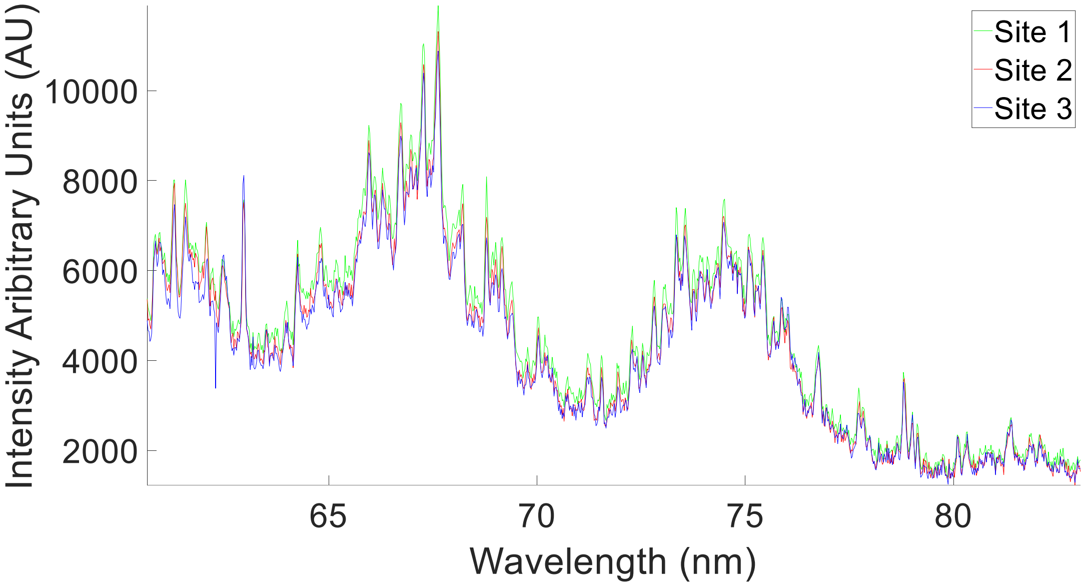


Figure E.3: Single shot spectra taken at three sites on a target surface located 1.50 mm from the optical axis of the spectrometer.

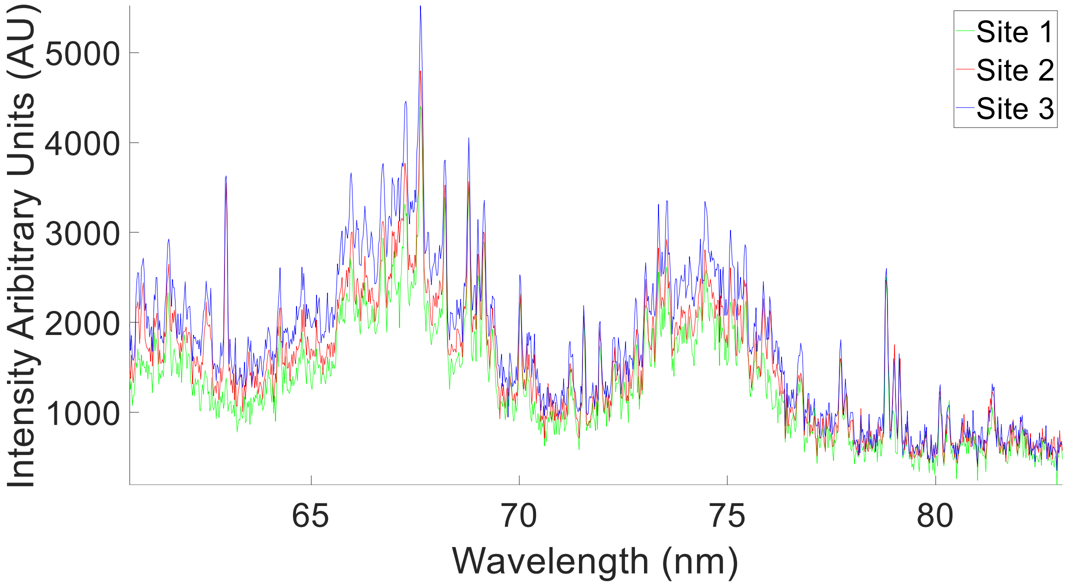


Figure E.4: Single shot spectra taken at three sites on a target surface located 1.75 mm from the optical axis of the spectrometer.

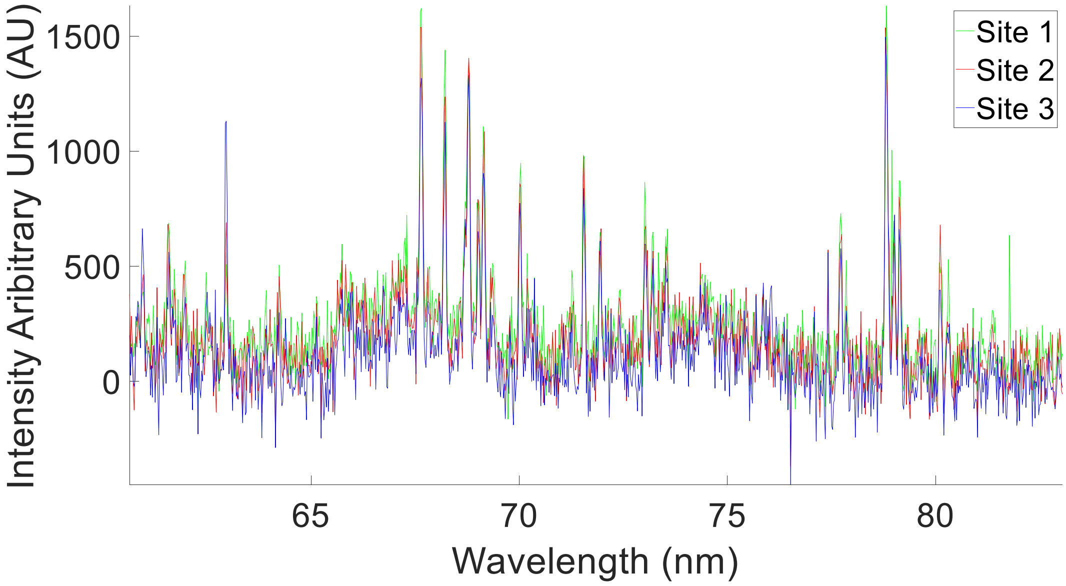


Figure E.5: Single shot spectra taken at three sites on a target surface located 2.00 mm from the optical axis of the spectrometer.

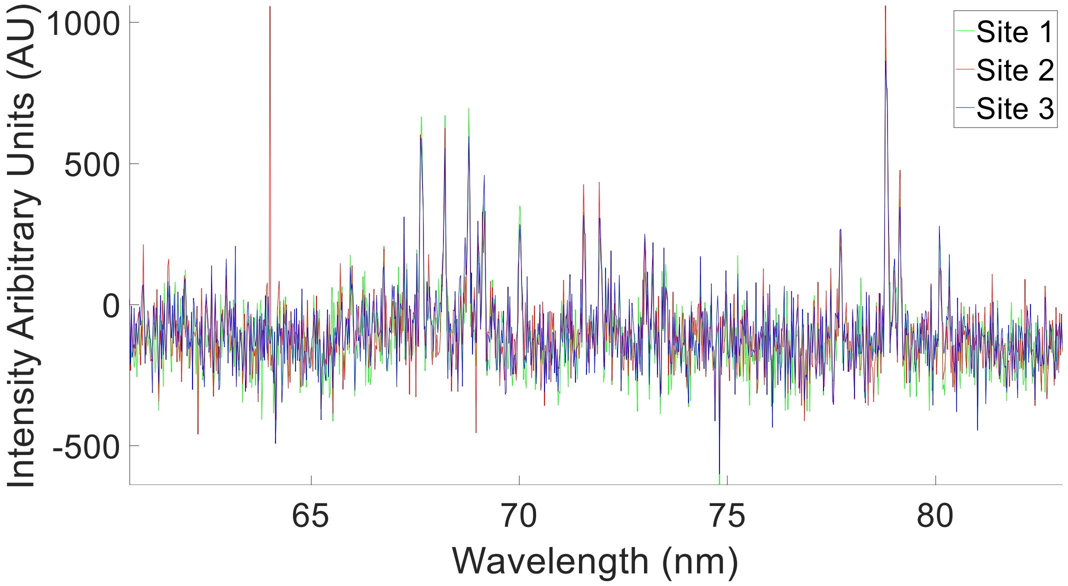


Figure E.6: Single shot spectra taken at three sites on a target surface located 2.25 mm from the optical axis of the spectrometer.