



Faculty of Science and Technology  
Department of Electrical Engineering and Computer Science

# Application of Machine Vision and Spectral Analysis in the Seafood industry

Master's Thesis in Computer Science  
by

Malavika Ramakrishnan

Internal Supervisors

Prof. Tomasz Wiktorski

External Supervisors

Dr. Helene Seyr

Stein-Kato Lindberg


June 15, 2021



# Declaration of Authorship

I, Malavika Ramakrishnan, declare that this thesis titled, ‘Application of Machine Vision and Spectral Analysis in the Seafood Industry’ and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a master’s degree at this University.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.

Signed:  \_\_\_\_\_

Date: 15 June, 2021 \_\_\_\_\_



*"The most powerful works of science fiction don't describe the future; they change it."*

Annalee Newitz

*"If you want to shine like the sun, first burn like the sun."*

Dr. A.P.J. Abdul Kalam



# *Abstract*

The Norwegian seafood industry contributes largely to the global economy as Norway is one of the leading producers of seafood. Atlantic salmon is a widely consumed species of fish and has a large global market. The nutritional value of salmon is affected by its quality and the demand to evaluate it is a problem whose solution is a work in progress. Earlier, seafood quality evaluation used to be laborious, time consuming, and was done using invasive procedures. This has changed in the recent past and hyperspectral imaging has become a popular non-invasive procedure for quality evaluation.

Among the many factors contributing to the the quality determination of fish, the presence of blood after filleting is in some markets highly undesirable. Hyperspectral image analysis on cod fish has proven to be highly successful for blood detection, but application of the same technique on salmon is a more difficult problem as it contains pigments whose spectral properties are similar to that of blood and interferes in the blood analysis.

This study focuses on a novel approach to detecting blood in salmon fillets using hyperspectral image analysis. A comparative study was performed on two types of data, spectral and abundance data, extracted from the hyperspectral images of salmon fillets to predict the concentration of blood in each pixel using different machine learning models. A regression analysis using each model on both types of data was conducted to find the appropriate data and the model that produced relatively better results.

Out of the 3 linear models, 2 non-parametric models, 2 ensemble models and 1 neural model applied on both types of data for the regression analysis, the gradient boosting model performed best on the raw spectral data while the linear regression performed the best on the abundance data. The raw spectral data was found to be better than the abundance data when the gradient boosting was applied on it to predict the blood concentration when considering the evaluation metrics. A modified version of this model was then integrated into the Maritech eye product, which is a hyperspectral imaging setup to enable automatic quality evaluation of salmon fillets.

**Keywords:** Hyperspectral analysis, salmon, blood detection, machine learning, regression analysis, Maritech Eye.





# *Acknowledgements*

This project in collaboration with Maritech Systems and Nofima, is a dissertation submitted in partial fulfilment of the requirements for the award of Master of Science in Computer Science at the University of Stavanger, Norway.

I would like to take this opportunity to thank everyone who have supported me throughout this journey and made this thesis possible. Prof Tomasz Wiktorski at the University of Stavanger is my main supervisor on this project and has been encouraging and supportive of my work for almost a year now. I am very grateful to you for believing in me and for all your valuable guidance and timely help for you have always striven to make sure this project was a success.

I wish to express my deepest gratitude to the CEO of Maritech Systems, Mr Odd Arne Kristengård and the VP Data Science, Dr Oddvar Husby for entrusting me with this project and having been magnanimous in providing me with all the resources I needed throughout this semester. You have been very down to earth and made me comfortable in these past months. I am very grateful to you for making me a part of the Maritech family.

Dr Helene Seyr is my external supervisor at Maritech and I cannot thank you enough for always looking out for me both professionally and personally. Your amazing insights and attention to details have made both me and this thesis so much better. You have always helped me out with the programming, brain-storming for ideas and the writing and have taught me to appreciate the good things while also being prepared for the worst. Through the duration of this thesis, I have gained you as a great mentor, an honest critic, an amazing friend and a caring sister. Thank you, again.

Stein-Kato Lindberg, scientist at Nofima is my external supervisor and a great pillar of support to this whole project. You have always been by my side (virtually, of course) and made sure I understand all the technical details. I am ever grateful for your patience in answering all my doubts related to fish, salmon, optics and hyperspectral imagery, for I know so much more about salmon now than I did at the beginning of this project, thanks to you. I am grateful to you for always guiding me and correcting me graciously and helping me grow.

I am very thankful to the Data Science team at Maritech, Luiza Oancea, Aleksander Kringstad, Birgitte Fidjeland, Ting Chao, Yngvild Neset, Kristoffer Heggdal and Anne Marie Mathisen, the marketing head of Maritech, Marie Gjære Gundersen for her kind words of support and everyone at Maritech for constantly supporting me and making me feel at home.

I would also like to thank the team at Nofima, Tromsø who have helped with the data acquisition for this project. Stein-Kato Lindberg and his crew including Stein Harris Olsen, the scientist who performed the chemical analysis, Torbjørn Tobiassen, scientist, Tatiana Ageeva, scientist, Margrethe Esaiassen, guest scientist and Iver Hovstad Røpstad, intern who all contributed to the filleting and manual sampling. Thank you, all. A special thanks to Mowi for providing us with samples of Atlantic Salmon, farmed and harvested at Buksevik and delivered at Nofima for data acquisition.

I thank my family and friends for believing in me and supporting me both in my professional and personal decisions and always encouraging me to achieve my goals. My mummy and appa who have been more of my best friends. My daddy thatha, Jai maam and Annaswamy thatha who would be so proud of me and might be smiling from above. My Susee paati for always encouraging me and my Babul for inspiring me to be strong, brave and an independent woman. I am also grateful to my parents-in law for being supportive and always praying for my well-being.

Special thanks to my sister, Jannu who has always been there for me, listened to me, advised me on critical life decisions and always counting on me. I can never thank you enough for everything you do for me, my ray of hope.

Finally, I would like to thank my husband Agaraoli, who is always proud of me and helped me out both professionally and personally. I look up to you and admire you so much, thank you for always having my back and being there for me.

# Contents

<b>Declaration of Authorship</b>	<b>iii</b>
<b>Abstract</b>	<b>vii</b>
<b>Acknowledgements</b>	<b>ix</b>
<b>Abbreviations</b>	<b>xv</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Hyperspectral Imaging . . . . .	2
1.3 Objective . . . . .	4
1.4 Structure of the thesis . . . . .	4
<b>2 Theoretical Background</b>	<b>7</b>
2.1 Hyperspectral Imaging . . . . .	7
2.1.1 Basic Perspective . . . . .	7
2.1.2 Operational Principle . . . . .	8
2.2 Optical Properties . . . . .	9
2.2.1 Beer-Lambert's law . . . . .	10
2.3 Analysis, Algorithms and Applications . . . . .	12
2.3.1 Glossary . . . . .	13
2.3.2 Spectral Unmixing . . . . .	14
2.3.3 Endmember spectra . . . . .	16
2.4 Previous Works . . . . .	17
2.4.1 Geometric vs Statistical Unmixing Algorithms . . . . .	17
2.4.2 Constrained Spectral Unmixing . . . . .	19
2.4.3 Ridge or centreline detection . . . . .	23
2.5 Regression Models . . . . .	25
2.5.1 Linear model . . . . .	26
2.5.2 Non-parametric model . . . . .	28
2.5.3 Ensemble model . . . . .	30
2.5.4 Neural Network model . . . . .	32

---

<b>3</b>	<b>Methodology</b>	<b>35</b>
3.1	Outline	35
3.2	Proposed Solution	36
3.2.1	Data Collection & Preparation	36
3.2.2	Data Pre-processing	37
3.2.3	Feature Engineering	38
3.2.4	Machine Learning Modelling	39
3.2.5	Product Integration	40
3.2.6	Summary	40
<b>4</b>	<b>Research Data and Analysis</b>	<b>43</b>
4.1	Data Collection And Preparation	43
4.1.1	Introduction	43
4.1.2	Hyperspectral Imaging Setup	44
4.1.3	Data Reconstruction	45
4.1.4	Sample collection	46
4.1.5	Chemical Analysis	47
4.2	Data Pre-processing	49
4.2.1	Determining Sampling locations	51
4.3	Feature Engineering	58
4.3.1	Spectral data	58
4.3.2	Abundance data	59
4.3.3	Outlier removal	60
<b>5</b>	<b>Model Development and Evaluation</b>	<b>61</b>
5.1	Technical Setup	61
5.2	Model Development	62
5.2.1	Parameter Tuning	63
5.3	Model Validation	65
5.3.1	Regression Residuals	65
5.3.2	Goodness of fit	66
5.3.3	Cross-Validation	67
5.4	Model Evaluation	68
5.4.1	Evaluation metrics	68
<b>6</b>	<b>Results and Discussion</b>	<b>71</b>
6.1	Model-wise Results	71
6.1.1	Linear model	72
6.1.2	Non-parametric model	75
6.1.3	Ensemble model	78
6.1.4	Neural model	81
6.2	Overall Results	83
6.3	Interpretations	86
6.4	Product Integration	88
6.4.1	Blood Response Image	89
6.5	Limitations	91

---

<b>7</b>	<b>Conclusions and Future Directions</b>	<b>93</b>
7.1	Summary . . . . .	93
7.2	Recommendations for Further Work . . . . .	94
	<b>List of Figures</b>	<b>94</b>
	<b>List of Tables</b>	<b>97</b>
	<b>A Salmon Health Certificate</b>	<b>99</b>
	<b>Bibliography</b>	<b>101</b>



# Abbreviations

**2D** 2-Dimensional

**2-DWT** 2-Dimensional Wavelet Transform

**3D** 3-Dimensional

**ANC** Abundance Non-negativity Constraint

**ANN** Artificial Neural Network

**ASC** Abundance Sum to one Constraint

**BSOM** Bayesian Self Organizing Maps

**CCD** Charge Couple Device

**CSUM** Constrained Spectral Unmixing

**CV** Cross Validation

**DECA** Dependent Component Analysis

**DN** Digital Numbers

**DPFT** Dark-Point-Fixed Transform

**EMR** Electro-Magnetic Radiation

**EMSC** Extended Multiplicative Scatter Correction

**ENVI** ENvironment for Visualizing Images

**FCLSU** Fully Constrained Least Squares Unmixing

**FPFT** Fixed-Point-Free Transform

**GB** Gradient Boosting

**GBDT** Gradient Boosted Decision Trees

**GDP** Gross Domestic Product

**GMM** Gaussian Mixture Model

**HIS** Hyperspectral Imaging

**ICA** Independent Component Analysis

**ICA-AQA** Independent Component Analysis -based Abundance Quantification Algorithm

---

**ICE** Iterative **C**onstrained **E**ndmembers  
**KICA** Kernel **I**ndependent **C**omponent **A**nalysis  
**KNN** **K**-Nearest **N**eighbors  
**LOO-CV** Leave **O**ne **O**ut **C**ross **V**alidation  
**MAE** Mean **A**bsolute **E**rror  
**MaxNG** Maximization of **N**on-**G**aussianity  
**MGS** Modified **G**ram-**S**chmidt algorithm  
**ML** Machine **L**earning  
**MLP** Multi-**L**ayer **P**erceptron  
**MNF** Maximum **N**oise **F**raction  
**MSE** Mean **S**quared **E**rror  
**MVT** Minimum-**V**olume **T**ransform  
**NAPC** Noise-**A**dded **P**rincipal **C**omponents  
**NCLS** Non-negative **C**onstrained **L**east **S**quares  
**NIR** Near **I**nfra-**R**ed  
**NLLS** Non-**L**inear **L**east **S**quares  
**NMF** Nonnegative **M**atrix **F**actorization  
**NNLS** Non-**N**egativity-constrained **L**east **S**quares  
**OLS** Ordinary **L**east **S**quares  
**ONNX** Open **N**eural **N**etwork **E**xchange  
**ORASIS** Optical **R**eal-time **A**daptive **S**pectral **I**dentification **S**ystem  
**OSP** Orthogonal **S**ubspace **P**rojection  
**PCA** Principal **C**omponent **A**nalysis  
**PCR** Principal **C**omponent **R**egression  
**PLS** Partial **L**east **S**quares  
**PPI** Pixel **P**urity **I**ndex  
**RBF** Radial **B**asis **F**unction  
**RF** Random **F**orest  
**RGB** **R**ed, **G**reen and **B**lue  
**RMSE** Root **M**ean **S**quared **E**rror  
**SNV** Standard **N**ormal **V**ariate  
**SPICE** Sparsity **P**romoting **I**terative **C**onstrained **E**ndmembers  
**Spy** Spectral **P**ython



**SVM** Support **V**ector **M**achines

**SVR** support **V**ector **R**egression

**TSS** Total **S**um of **S**quares

**UCLS** Unconstrained **L**east **S**quares

**VIS** **VIS**ible

**VIS-NIR** **VIS**ible and **N**ear **I**nfra-**R**ed

**WLS** Weighted **L**east **S**quares

**WLSU** Weighted **L**east **S**quares **U**nmixing



# Chapter 1

## Introduction

Apart from the oil, gas and energy industries, Norway's seafood industry is the second largest contributor to the Gross Domestic Product (GDP) of the nation. Owing to the country's long coastline, with wide seabeds, and suitable climatic conditions, the seafood industry has been one of the oldest and most reliable industries here. With global sales in 2016 of over \$10 billion, Norwegian seafood exports contribute to almost 10% of the entire global seafood market [1].

Atlantic salmon (derived from Latin name 'Salmo salar') is by far the largest species to be exported both in terms of volume and value. One reason for the boom in the salmon market is the increasing increasing popularity of sushi around the world. In 2019, the volume of salmon exported amounted to 1.1 million tonnes, representing NOK 72.5 billion in value [2]. Other popular seafood for export include fishes like trout, fresh and frozen codfish, mackerel and shellfish like prawns and king crabs.

### 1.1 Motivation

Fish is a widely consumed food all over the world for its many nutritional benefits and the quality of fish is an important factor for the fish industry and consumers [3]. The need to evaluate the quality of the fish both consumed and exported starts at the very beginning stages of fish farming and the research around aquaculture and seafood quality has been on the rise over the past few decades.

Salmon being one of the major species of fish that is consumed and exported widely, determining the quality of salmon fillets is an on-going research area. It is believed that the red coloration of salmon fillets is an important product property appreciated by consumers. The flesh color should be deep red and evenly distributed along the fillet [4].

Many other factors contribute to the determination of the quality of salmon, out of which the presence of blood in salmon fillets plays a major role. The presence of blood in any meat or fish after filleting is generally not desirable. The blood on a salmon fillet blackens on smoking which makes the presence of blood spots in fresh and smoked Atlantic salmon undesirable, and thus have become a concern to the industry and consumers. Many blood detection techniques have been used over the years to detect traces of blood in meat and seafood. Studies were conducted to observe the effects of slaughter procedure on residual blood in the fillet [5] among many others. But these procedures were mostly invasive and required chemical experimentation of each individual salmon fillet and not a general solution.

The need to come up with non-invasive blood detection techniques have been successfully satisfied by employing hyperspectral imaging. Detection of blood in white fish like cod has been done successfully using hyperspectral imaging techniques [6]. A similar approach to detecting blood in salmon fillets using hyperspectral image analysis is going to be explored in this thesis.

So far, the process of sorting the fish based on quality has been done manually, and it requires a lot of experienced labor. It is hard to judge the quality of salmon fillets before smoking, so a lot of smoked salmon gets discarded because of blood spots that they see after they cut it into slices. The main motivation for this thesis is to automate this process which can in turn be included in an automated production line. This increases the production efficiency without having to increase the number of experts and is more sustainable. In addition, we hope that the automatic detection is more uniform and reliable than manual sorting.

## 1.2 Hyperspectral Imaging

The electromagnetic spectrum is a term used to represent all types of electromagnetic radiation (EMR) including radio waves, X-rays, visible light, and infrared radiation. Electromagnetic radiation is the energy transmitted through empty space or through matter as electromagnetic waves. These are waves of different frequency and wavelength depending on their origin.

Frequency is the total number of occurrence of oscillations of a wave in a unit time. While wavelength is the distance the wave travels during one full cycle. Frequency is measured in hertz(Hz) and wavelength is measured in meter(m). Both of them are connected with

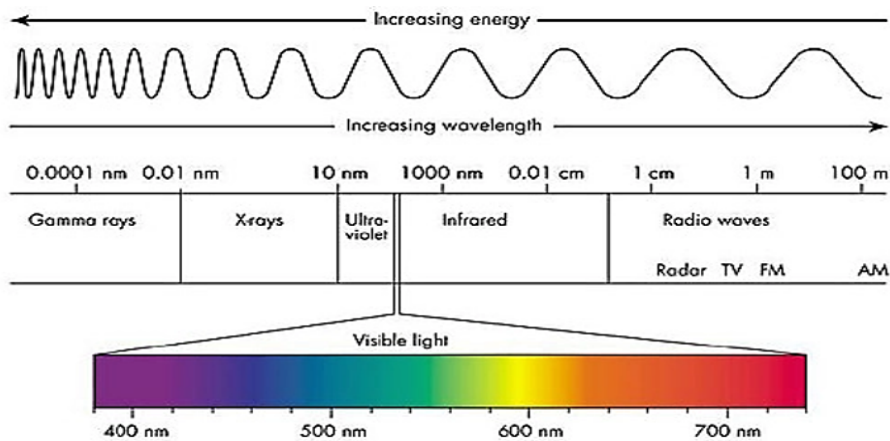
the equation:

$$c = \lambda \times f \quad (1.1)$$

where  $c$  is the speed of light,  $\lambda$  is the wavelength and  $f$  is the frequency. It can be seen from equation 1.1, that frequency and wavelength are inversely proportional, meaning, as frequency increases, wavelength decreases and vice versa.

The electromagnetic spectrum has frequencies ranging from below one hertz to above  $10^{25}$  hertz, corresponding to wavelengths from thousands of kilometers down to a fraction of the size of a sub-atomic particle. Figure 1.1 shows an illustration of the electromagnetic spectrum with their wavelengths and their corresponding applications. This spectrum is divided into separate bands and the waves in each of these bands have different characteristics. The study of interactions of the electromagnetic waves with matter is known as spectroscopy.

**Figure 1.1:** An illustration of the electromagnetic spectrum[7]



Electromagnetic radiation with a wavelength between 380 nm and 760 nm is known as visible light as it is the part of the electromagnetic spectrum the human eye is most sensitive to. A rainbow shows the visible part of the electromagnetic spectrum; infrared is located just beyond the red side of the rainbow with ultraviolet appearing just beyond the violet end and hence their names.

A normal imaging camera captures the information of an object from the light reflected by it in the visible region in the three wavelength bands most sensitive to the human eye, long wavelengths - perceived as red, medium wavelengths - perceived as green, and short wavelengths - perceived as blue.

Hyperspectral Imaging (HSI) unlike the usual imaging technique, captures more than just the RGB (Red, Green and Blue) information visible to the naked eye, it divides the

spectrum into multiple narrow bands over different wavelength ranges. This technique of dividing images into bands can be extended beyond the visible light region to the infrared and ultraviolet regions. The major regions of focus for our thesis are the Visible (VIS) and parts of the Near Infra-red (NIR) regions spanning across 400 nm to 1000 nm and is also called the VIS-NIR or Visible and Near Infra-Red region. It has been experimentally found that this is the region most susceptible to show variations in the optical properties of the chemical constituents of a fish, which is the focus of this thesis.

### 1.3 Objective

Detecting the amount blood in fish muscles is an important way of assessing the quality of a fish. In general, it can be assumed that the amount of blood present in the fish muscles after filleting is inversely proportional to its quality. In the case of Salmon, colour determines an additional desirable quality. The stronger the colour of Salmon, the more desirable it is, which makes it quite a challenge to differentiate the reddish pink pigment of the salmon fillet from blood spots.

Using hyperspectral imaging technique to detect blood in white-fish was done in the work [Skjelvareid et al. \[6\]](#). A hyperspectral camera picks up the differences in absorbance between blood and fish muscle of a cod fish as it is quite translucent, sort of like a fiber optic cable. It can be very efficient in transmitting light, which is why the interactance setup mentioned in the work [Skjelvareid et al. \[6\]](#) works. The absence of pigments in cod fish that interfere with the absorptive properties of blood makes it easier.

Whereas in Salmon (red-fish), it is complicated to separate the light absorption due to the blood from the light absorption due to the pigments of the red-fish since the optical properties of blood is often similar to that of the pigments in the red muscles.

This thesis introduces a novel approach in which hyperspectral analysis can be applied on the images of the salmon fillets to detect the presence of blood and its concentration. The model(s) developed in this thesis are to be integrated in the product Maritech Eye <sup>1</sup> which will help classify in real-time if a salmon fillet is of good quality (bloodless) or not (bloody).

### 1.4 Structure of the thesis

This thesis is organised as 7 chapters, each explaining in detail the steps involved. In Chapter 2, the theory for the thesis which includes the principles of hyperspectral imaging,

---

<sup>1</sup>Product link: [Maritech Eye](#)

---

optical properties, algorithms and machine learning models are discussed in detail along with the related works. Chapter 3 presents the methods and steps involved in detecting the presence of blood concentration in salmon fillets. An essential step of this thesis is the data pre-processing and data analysis which is presented in detail in Chapter 4. Chapter 5 presents in detail the experimental setup with the discussion of model development, validation and evaluation. This is followed by Chapter 6 which explains in detail the results from the eight models for both the spectral and abundance frameworks, and discusses the process of product integration and the limitations. Finally, Chapter 7 presents the conclusion and possible future directions.





## Chapter 2

# Theoretical Background

This chapter represents the theory behind this thesis and discusses in detail, the related works. The principles of hyperspectral imaging are explained in section 2.1, the optical properties are explained in 2.2, the related algorithms and analysis are described in 2.3, the relevant previous works have been discussed in section 2.4 and the machine learning models used in this thesis are discussed briefly in section 2.5.

### 2.1 Hyperspectral Imaging

As presented in section 1.2, hyperspectral imaging collects information from a number of electromagnetic spectral bands, over multiple wavelengths and captures the underlying processes like the chemical characteristics and biophysical properties at the pixel level. Every pixel of a hyperspectral image consists of multiple layers of spectral information about the components of substances present in them. Detecting these substances is possible due to their different optical properties to different wavelengths of the electromagnetic spectrum.

#### 2.1.1 Basic Perspective

Hyperspectral imaging can be thought of as a spectroscopic procedure. Spectroscopy is the technique used in determining the chemical composition of an object by means of light and gives a detailed fingerprint from the interaction between the electromagnetic spectra and the atoms and molecules. They provide both qualitative and quantitative measurements of the chemical components of an object. This can be measured in transmittance, reflectance, absorbance, phosphorescence, fluorescence and/or radioactive decay [8].

Just like a normal digital camera captures data in the visible light region over the three wavelength bands (RGB) visible to the naked eye, Hyperspectral imaging (HSI) captures data in a contiguous range of multiple wavelengths over the whole electromagnetic spectrum over narrow bands. It may use the infrared, the visible spectrum, the ultraviolet, x-rays, or some combination of the above.

Mathematically, an image is represented as a 3-Dimensional (3D) array where the  $x$  and  $y$  co-ordinates contain the spatial information of every pixel while the  $z$  co-ordinates contain the colour specific pixel information whereas in a hyperspectral image, the  $z$  co-ordinates contain the spectral pixel information in terms of number of bands or rather the amount of light for every wavelength in which data was captured. In other words, the spectral information contained in each pixel of a HSI image forms the third-dimension to an otherwise 2-Dimensional (2D) spatial image in turn generating a hypercube or an image cube.

**Figure 2.1:** A hypercube vs RGB image [9]

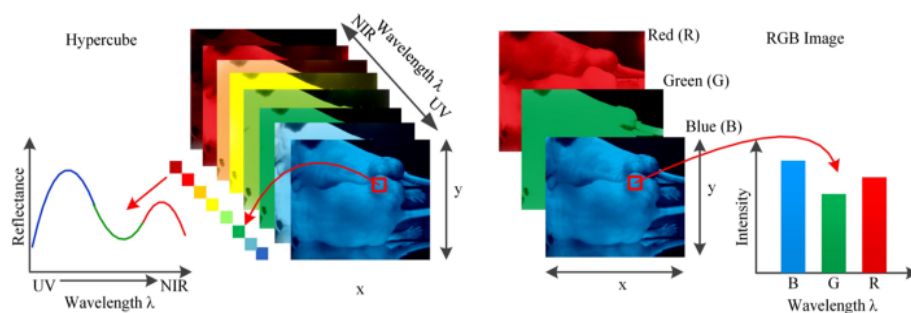


Figure 2.1 shows an example representation of a hypercube against an RGB image of the same object. It can be seen that a hypercube consists of multiple layers over a range of wavelength bands while the RGB image has data in only the three wavelength bands. This hyperspectral data includes absorption spectrum, reflectance spectrum or emission spectrum of the materials present in each pixel.

### 2.1.2 Operational Principle

Capturing a hyperspectral image is a process in which the light from the source is focused on the object through a slit followed by a dispersive element like a prism or diffraction grating. This splits the white light or rather a combination of different wavelengths, into multiple wavelength bands. This dispersive element is followed by an objective lens which focuses the light onto a charge couple device (CCD). The CCD is the detecting array where the amount of incoming light is measured.

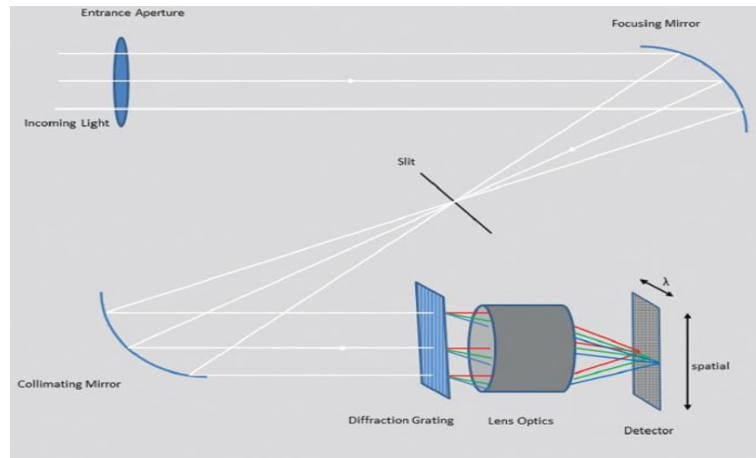
**Figure 2.2:** Operating principle of a hyperspectral camera [10]

Figure 2.2 shows the hyperspectral imaging measurement system from Norsk Elektro Optikk.2. Considering the 'pushbroom' <sup>1</sup> camera which is used in this study. When the light is focused on the object to be scanned, a hyperspectral camera in pushbroom mode captures the light reflected back from the object sequentially. It captures data one line at a time meaning it collects the spectral data from a single spatial dimension at a time resulting in an image layer with one dimension of spatial data and the other spectral data. These layers are stacked up one on top of the other to give a 2-D spatial image with the third dimension containing the spectral data from each layer.

## 2.2 Optical Properties

The optical properties of an object describe how its molecules interact with light. Let us consider when electromagnetic radiation (or light) of any wavelength is incident on an object. Some of the light goes right through it unchanged, this is called *transmittance* of the object. Some of the radiation changes direction on hitting the object but still has the same energy, this is called *scattering*. Some part of the radiation gets absorbed by the object after hitting it and is called *absorption*. And some part of the radiation which is reflected off the object is called *reflectance*. Some other optical properties of matter include refraction, polarization, diffraction, and dispersion.

The properties affecting the spectra that I have focused on in this thesis are absorption, reflectance, transmittance and scattering. Transmittance is calculated as the fraction of

<sup>1</sup>Pushbroom method in hyperspectral cameras capture images line by line, sequentially and results in a 2d image for each spectral channel.

incident (input) light to that of the transmitted (output) light.

$$T = \frac{I}{I_o} \quad (2.1)$$

where  $T$  is the transmittance,  $I$  is the transmitted light and  $I_o$  is the incident light. Absorbance is calculated as a logarithmic function of transmittance.

$$A = \log_{10} \frac{1}{T} = -\log_{10}(T) \quad (2.2)$$

where  $A$  is the absorbance and  $T$  is the transmittance. Reflectance is calculated from Absorbance and Transmittance as:

$$R = 1 - A - T \quad (2.3)$$

where  $R$  is the reflectance,  $A$  is the absorbance and  $T$  is the transmittance.

The amount of absorption and scattering vary according to the incident wavelength band and depend on the chemical composition and structure of the object. Some objects will absorb certain wavelengths of light while some reflect some wavelengths. A reflectance or absorbance spectrum <sup>2</sup> of an object can help identify the chemical composition of the object. An object containing several materials or a mixture of materials will have a complex spectra and thus their corresponding spectral information can be used to determine the composition of the mixture of materials.

### 2.2.1 Beer-Lambert's law

This thesis considers a salmon fillet whose composition is known to be a mixture of fat, water, blood and proteins. The idea here is to calculate the absorbance and thus calculate the composition of the mixture of materials present in our fillet.

This can be done using the **Beer-Lambert's law**. It is a combination of the Lambert's law <sup>3</sup> and Beer's law <sup>4</sup> and connects the absorbance to both the concentrations of the object and the thickness of the material sample [11].

This can be mathematically expressed as:

$$A \propto C \times B \quad (2.4)$$

---

<sup>2</sup>A reflectance or absorbance spectrum shows the reflectance or absorbance respectively of a material measured across a range of wavelengths.

<sup>3</sup>Lambert's law states that the loss of light intensity when it propagates in a medium is directly proportional to the intensity and path length.

<sup>4</sup>Beer's law states that the transmittance of a solution remains constant if the product of concentration and path length stays constant.

where  $A$  is the absorbance,  $C$  stands for Concentration and  $B$  stands for Thickness. This proportionality can be converted into an equality by including a proportionality constant ( $\varepsilon$ ) which is the molar absorption coefficient or absorptivity.

$$A = (\varepsilon) \times C \times B \quad (2.5)$$

where  $\varepsilon$  is the proportionality constant Absorptivity. This can also be written as:

$$\log_{10} \frac{I_o}{I} = (\varepsilon) \times (C) \times (B) \quad (2.6)$$

where  $I_o$  is the incident light,  $I$  is the transmitted light,  $\varepsilon$  is the molar absorptivity (in units of moles per liter),  $C$  is the analyte concentration (in gram per 100 ml of the solution) of the object and  $B$  is the path length [12].

It must be noted that the transmitted light is measured as the light not lost from the incident beam of light or the light that makes it through the sample to the detector. The lost part is expected to describe the absorptive properties of the object but when a considerable amount of light incident on the object is lost due to scattering, the whole calculation becomes difficult since the lost light cannot be categorized for sure as either absorption or scattering.

Assuming an ideally absorbing medium or a perfect absorber with no scattering, Beer's law for ideally absorbing media can be stated as:

$$I(x) = I_o e^{-\mu_a x} \quad (2.7)$$

where  $I$  is the intensity,  $x$  is the distance in the direction of propagation,  $I_o$  is the intensity at distance  $x = 0$ , and  $\mu_a$  is the absorption coefficient.

Similarly, assuming an ideally scattering medium or a perfect scatterer with no absorption, Beer's law for ideally scattering media can be stated as:

$$I(x) = I_o e^{-\mu_s x} \quad (2.8)$$

where  $I$  is the intensity,  $x$  is the distance in the direction of propagation,  $I_o$  is the intensity at distance  $x = 0$ , and  $\mu_s$  is the scattering coefficient.

Now, combining the above two laws of absorption and scattering, we have

$$I(x) = I_o e^{-\mu_t x} \quad (2.9)$$

where  $I$  is the intensity,  $x$  is the distance in the direction of propagation,  $I_o$  is the intensity at distance  $x = 0$ , and  $\mu_t$  is the extinction coefficient,  $\mu_t = \mu_s + \mu_a$ .

Thus, Beer's law describes an exponential relationship between the intensity output and the extinction coefficient (a combination of scattering and absorption) and distance propagated through the object [8].

## 2.3 Analysis, Algorithms and Applications

Hyperspectral data is usually used in the field of remote sensing for robust characterization of target and background signatures and scene characterization. Few other techniques include anomaly detection, target detection and classification, dimensionality reduction and reconstruction and can be applied to many other fields like astronomy, agriculture, food technology, molecular biology, biomedical imaging, geosciences, physics, and surveillance. There are several spectral software libraries that can be used to do this analysis, like Environment for Visualizing Images (ENVI) <sup>5</sup> and Spectral Python (SPy) <sup>6</sup>[13].

The spectral information present in every pixel of a HSI can indicate the presence of different materials in it. Some constituent materials could be smaller than the pixel itself. It is important to understand the **Pixel Purity Index (PPI)** as it can help understand the spectral information in hyperspectral data better. A *pure pixel* is one which has only one constituting material whereas a *mixed pixel* is one which has multiple materials or a mixture of multiple materials at varying concentrations. Mixed pixels result from homogeneous mixtures of distinct materials.

HSI analysis algorithms are predominantly of two types based on whether the entire pixel is used or a part of the pixel for analysis. **Whole pixel methods** are used to identify one or more target materials in every pixel based on the spectral similarity between the pixel and target spectra. **Sub-pixel methods** can be used to identify not just the presence of the target materials but also the concentration in which it is present in each and every pixel.

One common method of analysis is to compare the spectral data from the pixel with their corresponding reference spectra, also called a target spectrum to be able to identify each individual constituent material in a pixel. A good example would be a HSI from remote sensing like figure 2.3 where the image captures a region of land consisting of soil, water and minerals in varying distributions. Each pixel in such an image has one or more of the substances mentioned in different amounts and their corresponding spectra are called target spectra which can be used to identify the presence and concentration of the chemical constituents of a sample.

---

<sup>5</sup>ENVI: [Environment for Visualizing Images](#)

<sup>6</sup>SPy: [Spectral Python](#)

### 2.3.1 Glossary

Every pixel in a HSI of a sample substance can be thought to be made up of multiple layers of different characteristic information about the materials present in them. The surface of a sample has a small number of distinct materials that have relatively constant spectral properties.

Some formal definitions according to [14] are as follows:

- **Endmember** : a set of macroscopically pure spectral components present homogeneously or heterogeneously mixed in a sample.
- **Abundances** : constituting concentration or fractional coverage of elements present in a sample.

The individual constituent substances present in every pixel are called **endmembers** and the concentrations in which they are present are referred to as **abundances**.

Every endmember has its own spectral signature which can be used to detect them in a pixel using an endmember extraction/detection algorithm and the concentration in which they are present can be found using an abundance estimation algorithm.

Properties of abundances and endmembers to bear in mind:

- The spectra of any and all endmembers are non-negative. This implies that a negative reflectance is not possible.
- The abundances of an endmember are non-negative. This is also called the **Abundance Non-negativity Constraint (ANC)**
- The abundances for each pixel always sum to one. This is also called the **Abundance Sum to one Constraint (ASC)**

It is to be noted that the ANC and ASC are optional constraints in the calculations, while the non-negative reflectance requirement is a physical property of light.

#### **Spectral Mixture:**

Spectral mixture is the combination of spectral signatures of the various endmembers present in a pixel of a hyperspectral image. There are two main types of spectral mixtures according to [14]

- **Linear Spectral mixture**

- Linear spectral mixture is one in which the pixels appear in a spatially separated pattern like a square checkerboard.
- That is, the spectrum of a mixed pixel is a linear combination of the endmember spectra weighted by the fractional area coverage of each endmember in a pixel.
- The mixing systematics between these components are linear.

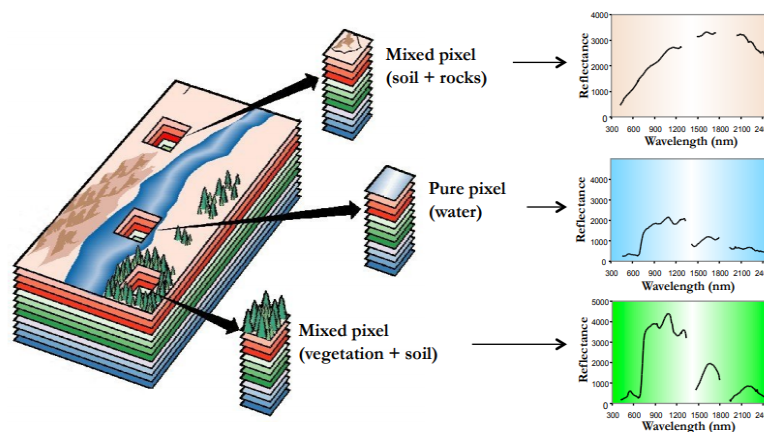
- **Non - Linear Spectral mixture**

- Non - Linear spectral mixture is one in which the pixels appear as a homogeneous mixture like grains of sand on a beach.
- That is, the endmember components are randomly distributed throughout the surface and are hard to distinguish from one another.
- The mixing systematics between these different components are nonlinear.

### 2.3.2 Spectral Unmixing

Spectral unmixing is the process of estimating the relative abundance of the endmembers in each mixed pixel by decomposing or 'unmixing' each mixed pixel down to its endmember spectra and their corresponding abundances. Figure 2.3 shows an illustration of Spectral Unmixing of mixed pixels.

**Figure 2.3:** An illustration of Spectral Unmixing [14]



It can be seen that spectral unmixing is necessary to understand the endmember distribution in a mixed pixel. Spectral unmixing is done in a series of steps like dimensionality reduction, endmember extraction and abundance estimation which are explained in detail below.



## **Dimensionality Reduction**

Dimensionality reduction is a popular pre-processing step used to enhance the computing performance of the unmixing algorithm especially on noisy data. Dimensionality reduction algorithms used for unmixing do not focus on reducing the dimension of the data with the goal of possible reconstruction of the reduced data to the original data. Instead, they focus on reducing the data to its minimal representation in a lower dimensional space that retains sufficient information for successful unmixing in that lower dimension.

Non-statistical algorithm like Optical real-time adaptive spectral identification system (ORASIS) and statistical algorithms like Principal Component Analysis (PCA), Maximum noise fraction (MNF), Independent Component Analysis (ICA) and Noise-adjusted principal components (NAPC) are examples of dimensionality reduction algorithms.

## **Endmember Extraction**

Endmember extraction or determination is the process of identifying the constituent material or mixture of materials in a sample pixel. Endmembers usually retain physical characteristics of the constituent chemical substance like absorption bands and spectral intervals which can be useful during the unmixing process to find the presence or absence of a target substance.

Non-statistical algorithms like Dark-point-fixed transform (DPFT) and Fixed-point-free transform (FPFT) both of which are variants of Minimum-volume transform (MVT) algorithms assume that the endmembers are deterministic quantities. Whereas statistical algorithms view endmembers as either deterministic, with an associated degree of uncertainty, or as fully stochastic, with random variables having probability density functions. Examples of statistical algorithms include Pure Pixel Index (PPI), orthogonal subspace projection (OSP), Fuzzy K-Means and Non-linear Least Squares (NLLS).

## **Abundance Estimation**

Abundance estimation is the process of evaluating the quantity or concentration in which the endmembers are distributed in the sample. There could be cases when endmembers are unknown, in such cases, there exists a group of abundance estimation based on blind source separation, which does not require endmember signatures to be known. Other than those, the algorithms used to estimate the abundance of a known endmember are as follows:

- Unconstrained Least Squares (UCLS)
  - This method can be used when the number of endmembers and their spectral signatures are all known.
  - This is an unconstrained solution which does not satisfy the abundance non-negativity and the abundance sum-to-one constraints
  
- Non-negative constrained least squares (NCLS)
  - If the abundance non-negativity constraint needs to be satisfied, the problem of abundance estimation becomes a constrained optimization problem.
  - However, imposing the ANC constraint can significantly increase the computational complexity of the abundance estimation problem
  
- Fully constrained least squares unmixing (FCLSU)
  - If both the ANC (abundance non-negativity) and the ASC (abundance sum-to-one constraints) constraints need to be satisfied, the problem of abundance estimation becomes an even more complicated one
  
- Weighted least squares unmixing (WLSU)
  - When only partial endmember information is known, abundances can be estimated via the weighted least squares (WLS) solution.
  - For instance, we may know the foreground endmembers only, but the information of background endmembers is more difficult to be determined.

### 2.3.3 Endmember spectra

Endmember spectra or otherwise called target spectra is the spectrum of the individual chemical constituent of an object. Spectrum is the representation of the optical properties of a material as a function of the wavelengths it responds to. In the example discussed in section 2.3 with figure 2.3, soil, water and vegetation are the endmembers. Calculating the spectrum of such individual chemical component is done using **spectrophotometry**.

Spectrophotometry is a branch of electromagnetic spectroscopy which enables the quantitative measurement of the optical properties of a material as a function of their wavelength. Spectrophotometers are the instruments that can measure the intensity of a light ray at different wavelengths and it works by measuring the amount of light absorbed by a chemical substance.

The chemical substances whose spectrum is to be calculated are isolated to their purest form and are scanned under a spectrophotometer. The resulting transmittance or reflectance can be used to obtain their absorbance spectra which is what we refer to in this thesis as endmember spectra. The endmembers relevant to this thesis which are found in a salmon fillet are water, fat, muscle, pigments like astaxanthin, betacaroten and cantaxanthin, the three oxidative states of blood namely, oxy-haemoglobin, deoxy-haemoglobin and met-haemoglobin making up in total 9 individual endmembers. The values used in this thesis are pre-calculated values by the research scientists at Nofima, Tromsø <sup>7</sup> who are our research partners in this project. This is explained more in detail in section 4.1.5

## 2.4 Previous Works

Detecting the presence of blood in fish muscle has multiple different solutions over the years, each improving on top of the past ones. This section explains in detail a few related works which have been an inspiration, scientific basis and guidance for this thesis.

The basic concepts of spectral unmixing has been discussed in section 2.3.2. A detailed study into the taxonomy of the unmixing techniques have been done in [15] which is explained below. Followed by a novel blood detection in cod fish by Martin et.al, in [6]. And a ridge detection in cod fish by Sivertsen et.al, in [16] are all discussed in this section.

### 2.4.1 Geometric vs Statistical Unmixing Algorithms

**PAPER 1:** This paper by Mario et.al, [15] provides an overview of the existing techniques for endmember extraction and unmixing, with particular attention paid to the distinction between statistical and geometrical approaches for spectral unmixing.

It can be understood that a system of linear equations in which, given a set of pure spectral signatures (endmembers) the actual unmixing is to determine apparent pixel abundance fractions and can be defined in terms of a linear numerical inversion process. Although the linear model has practical advantages such as ease of implementation and flexibility in different applications, nonlinear spectral unmixing may best characterize the resultant mixed spectra for certain endmember distributions. In practice, the nonlinear model requires a priori knowledge about the optical properties of the observed objects and is less computationally tractable.

---

<sup>7</sup>Nofima

The two basic types of spectral unmixing detailed in this paper are Geometric and Statistical unmixing

**Geometric Unmixing** As discussed earlier in section 2.3.2, linear spectral unmixing assumes that the collected spectra at the spectrometer can be expressed in the form of a linear combination of endmembers weighted by their corresponding abundances. This definition fits well with the geometrical selection of endmembers from the vertices of an enclosed shape or simplex, a polyhedron or a convex cone that minimally encloses or is maximally contained in the data.

This paper explains that a simplex-shrinking algorithm tries to find the minimum-volume simplex, i.e. the one that embraces the data as tightly as possible whereas a simplex-growing algorithm tries to figure out the endmembers through searching for a simplex growing from within the data.

An example discussed in this paper for the simplex-shrinking algorithm is Optical real-time adaptive spectral identification system (ORASIS) which uses a modified Gram-Schmidt (MGS) algorithm to factor the data matrix and then a shrink wrapping technique to find an outer simplex that encloses the data.

This paper states that the N-FINDR is a simultaneous simplex-growing algorithm that finds pure pixels that can be used to describe the mixed pixels in the scene. The pixel purity index (PPI) is stated as a simultaneous endmember extraction algorithm that works by projecting each pixel onto one vector from a set of random vectors spanning the reflectance space. A pixel receives a score when it represent an extremum of all the projections. Pixels with the highest scores are deemed to be spectrally pure.

Another algorithm discussed in this paper is the Iterative constrained endmembers (ICE) along with its sparsity-promoting version SPICE fits a simplex to the data while penalizing the volume of such simplex. The ICE algorithm does need a dimension reduction step, performed by the minimum noise fraction (MNF) algorithm. Algorithms based on the nonnegative matrix factorization (NMF) try to find the cone or the convex polyhedron that best fits the data and identifies the vertices of such object as the endmember of the scene. This paper states that such methods do not require the presence of pure pixels.

**Statistical Unmixing** If a spectral unmixing algorithm processes a mixed pixel by using statistical representations, then the algorithm is essentially a statistical unmixing algorithm. This paper states that the representations can either be parametric, which is analytical expressions that represent probability density functions or non-parametric functions.

An example of parametric algorithms stated in this work is the stochastic mixing model, in which each endmember distribution has Gaussian probability density function. There are algorithms that can be used for both endmember extraction and abundance estimation and they work by decomposing each pixel as a linear combination of pure endmember spectra. The estimation is conducted by generating the posterior distribution of abundances and endmember parameters under a hierarchical Bayesian model. The method introduces Bayesian self organizing maps (BSOM) and combines them with Gaussian mixture model (GMM) to model the spectral mixtures.

Some nonparametric statistical unmixing approaches discussed in this paper propose variations on the independent component analysis (ICA) method, called ICA-based abundance quantification algorithm (ICA-AQA) which is a high-order statistics based technique, that can accomplish endmember extraction and abundance quantification simultaneously. Another new unmixing method called dependent component analysis (DECA) is found by the authors to model the abundance fractions as mixtures of Dirichlet densities, thus enforcing the constraints on abundance fractions, namely non-negativity and constant sum. Another kind of dependent component analysis approach for unmixing is also maximization of non-gaussianity (MaxNG)

Apart from the above, this paper states that support vector machines (SVM) have also been recently used for unmixing and a novel method which integrates two-dimensional wavelet transform (2-DWT) and kernel independent component analysis (KICA) technique has also been used for unmixing.

### 2.4.2 Constrained Spectral Unmixing

**PAPER 2:** One of the most intriguing efforts to detect blood in fish muscles is explained in [6], where constrained spectral unmixing method was applied to detect the presence of blood in cod (white-fish). The method is based on diffuse reflectance hyperspectral imaging in the VIS/NIR range (430-1000 nm), and unmixing of measured absorbance spectra into known spectra for hemoglobin, water and muscle tissue. This a statistical linear algorithm that emphasises on the Non-Negative Least Squares method to do the unmixing.

Absorption spectra for cod hemoglobin within the visible (VIS: 430-780 nm) part of the electromagnetic spectrum has been adopted by the authors. It should be noted that the absorption spectrum varies with the oxidative state of the blood. When a fish is alive, the blood contains a mix of oxy- and deoxy-haemoglobin, and after the fish is slaughtered and filleted, the hemoglobin gradually undergoes natural oxidation and

becomes met-haemoglobin. The absorption spectra for these hemoglobin states can be used to estimate blood concentration in hyperspectral images of fish fillets.

In this work, the authors have employed a method termed non-negativity constrained spectral unmixing, which decomposes the measured spectrum into a set of absorption spectra known as a priori. Since the absorption spectra of hemoglobin in all three oxidation states are well known, the technique has potential for detection of blood in fish fillets and this combination of hyperspectral imaging and spectral unmixing to estimate of blood in fish fillets that has been done by [Skjelvareid et al.](#) in [6] is novel.

The constrained spectral unmixing method is also combined with an estimation of effects caused by light scattering. Scattering in biological tissues is an active research field in itself, especially within the field of medicine. In general, the interaction between tissue and light is a complex function of the absorbance of different chromophores (e.g. blood) and the tissue scattering properties, and it is difficult to separate the two exactly in measurements. Wavelength-dependent scattering effects also introduce a wavelength-dependent optical path length in tissue, and modifications to the **Beer-Lambert law** have been proposed in [17] to include scattering and variable path length described in [18].

In this work, the authors have made a simple assumption that the variable path length effect is negligible, and that the effect of scattering is to introduce a shift in the measured absorbance spectra. The baseline is assumed to be a slowly varying function of wavelength and is modelled as a low-order polynomial.

This approach is similar to the extended multiplicative scatter correction (EMSC) often applied in infrared spectroscopy. **EMSC** or extended multiplicative scatter correction, is a powerful pre-processing technique that isolates and removes complicated multiplicative effects caused by physical phenomena so that chemical effects can be more easily modelled and has been discussed in detail in [19].

The underlying assumption of the spectral unmixing model used here is that the total chemical absorption of light is a sum of absorption spectra for the different constituents in the mixture, weighted by their concentrations, which is explained in the **additive mixing model** in [20].

The total chemical absorption of light is a sum of absorption spectra for the different constituents in the mixture, weighted by their concentrations.

$$s_{meas}(\lambda) = \sum_{i=1}^N a_i e_i(\lambda) + \sum_{i=1}^M b_i \lambda^{i-1} \quad (2.10)$$

$$\mathbf{s} = \mathbf{E}\mathbf{a} + \mathbf{\Lambda}\mathbf{b} \quad (2.11)$$

where  $\mathbf{s} = [s_1, s_1, \dots, s_K]^T$  or  $s_{meas}(\lambda)$  is the total measured spectrum measured at  $K$  discrete wavelengths  $[\lambda_1, \lambda_2, \dots, \lambda_K]$ ,

$\mathbf{E} = [\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_N]$  is the endmember spectra where the individual spectra are given by

$$\mathbf{e}_i = [e_{i,1}, e_{i,2}, \dots, e_{i,K}]^T,$$

the endmember abundances are represented by  $\mathbf{a} = [a_1, a_2, \dots, a_N]^T$ ,

the polynomial matrix of wavelengths contained in  $\Lambda = [\boldsymbol{\lambda}^0, \boldsymbol{\lambda}^1, \dots, \boldsymbol{\lambda}^{M-1}]$  where  $\boldsymbol{\lambda}^i = [\lambda_1^i, \lambda_2^i, \dots, \lambda_K^i]^T$

and  $\mathbf{b} = [b_1, b_2, \dots, b_M]^T$  represents the scattering coefficients.

Now, spectral unmixing of the measured spectrum  $s$  is performed to estimate the individual abundances  $a_i$  while simultaneously taking into account the effects of scattering. As stated in 2.3.1, the abundances cannot have a negative value, so  $a_i \geq 0$  for all  $i$ . The scattering coefficients  $b_i$  are not constrained in any way. Thus, eq. 2.11 poses a mixed constraint problem. This can be separated into purely constrained and unconstrained parts, an approach known as separable least squares discussed in [21].

**Separable least squares** are generally written in the form  $|\mathbf{y} - \mathbf{A}(\mathbf{q})\mathbf{c}|^2 = \min$  [22] where minimization has to be carried out with respect to the parameters  $\mathbf{q}$  and  $\mathbf{c}$  and the latter enter linearly into the expression of the objective function. This leads to considerable simplifications.

Eq. 2.11 can be written as:

$$\Lambda b = s - Ea \tag{2.12}$$

From the normal linear least squares solution, a best-fit estimate for  $b$  can be obtained as:

$$\hat{b} = (\Lambda^T \Lambda)^{-1} \Lambda^T (s - Ea) \tag{2.13}$$

where  $(\Lambda^T \Lambda)^{-1} \Lambda^T = \Lambda^+$  is called the pseudoinverse and  $\hat{b}$  is the estimate for  $b$ . When eq. 2.13 is inserted in eq. 2.11, it becomes,

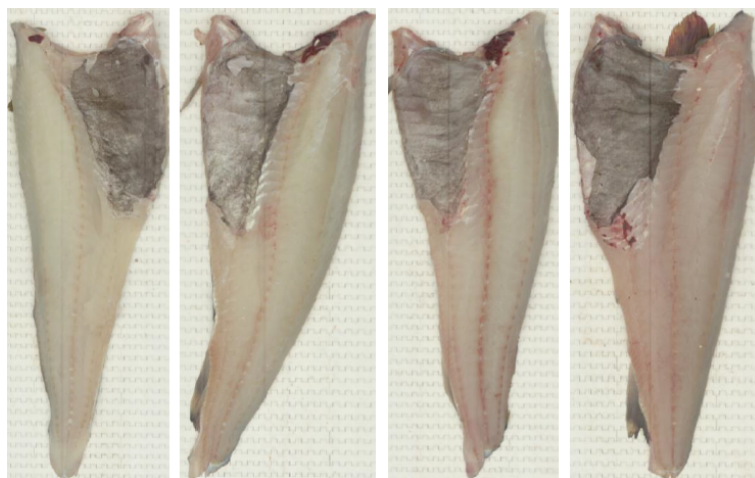
$$(I - \Lambda \Lambda^+) s = (I - \Lambda \Lambda^+) Ea \tag{2.14}$$

where  $(I - \Lambda \Lambda^+) = P$  is called the projection matrix which transforms the spectra to a vector subspace where the least-squares solutions for the scattering coefficients  $b$  are already applied.  $Ps = \tilde{s}$  and  $PE = \tilde{E}$  resulting is a purely constrained problem,

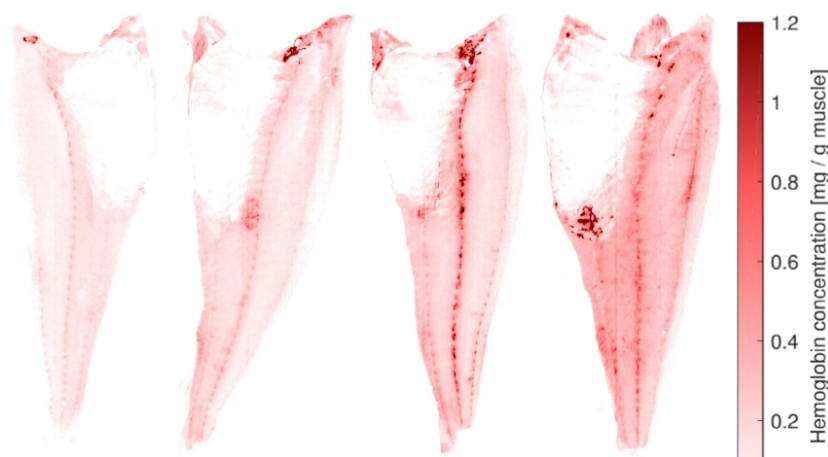
$$\tilde{s} = \tilde{E}a, \quad a \geq 0 \tag{2.15}$$

which can be solved with an algorithm adapted for non-negativity-constrained least squares (NNLS) problems explained in [23]. Solving Eq. 2.15 yields a least-squares solution  $\hat{a}$  which is inserted into Eq. 2.13 to obtain the scattering coefficients  $\hat{b}$ .

On solving the above equations, the scatter corrected spectrum of the endmembers is obtained by the authors, which is then modelled as a sum of blood, water and muscle endmembers. The three states of haemoglobin are summed together to be considered a single entity blood along with pigments like astaxanthin, betacaroten, cantaxanthin and muscle spectra are compared with the measured spectra of the HSI image of the codfish to develop a bloodmap. This bloodmap shows the intensity or concentration of blood in different regions of the fish.



(a) Colour images of cod fillets



(b) Blood concentration images of cod fillets

**Figure 2.4:** Color images (a) and corresponding blood concentration images (b) from four example fillets [6]

A calibration curve is presented by Skjelvareid et al. in [6] which they have used to convert the blood abundancies to the blood concentrations. Figure 6.21 shows the colour



images and their corresponding blood response images after applying the **Constrained Spectral Unmixing** (CSUM) technique.

### 2.4.3 Ridge or centreline detection

**PAPER 3:** One important step in automatic fish fillet inspection algorithms is segmentation, which can be used to label different regions in the image. For cod fillets, this can be applied to identify which part of the fillet belongs to the loin, belly flap, centre cut and tail. The severity of a flaw depends on the part of the fillet it is located in. The centreline, consisting of veins and arteries cut off during filleting, is always visible on cod fillets and hence a good reference for segmentation.

The work by Sivertsen et.al in [16], describes a process to enhance the centreline by using the absorption characteristics of haemoglobin, and how a novel ridge detection method can detect the centreline in cod fillets. This method enables both spatial and spectral identification of irregularities in the muscle, and makes it possible to extract information regarding the chemical composition of objects or areas in the image, such as blood, fat or water content. The centreline is enhanced by extracting a small subset of the hyperspectral data using the absorption characteristic of haemoglobin.

In this study, illumination and measurements were performed on the same side of the sample during interactance; the illumination was however focused on an area adjacent and parallel to the detectors' field of view. Keeping the angle of illumination and measurement parallel was important to have a constant optical path length, with varying sample thickness. It should be noted that the autoxidation rate of haemoglobin increases with storage temperature and varies between species and hence the authors suggest that the colour of the centre line to get darker as the fish fillet is stored.

In this work, to discriminate the ridge from the muscles, they attempted to increase the signal of the ridge while simultaneously decreasing the signal of the surrounding muscles which will highlight the ridge. This was done by performing numerical divisions between two different wavelength bands chosen in such a way that the absorbance wavelengths of the ridge and surrounding muscles had the highest contrast and variance. The ridge enhanced image was calculated as follows:

$$I_r = \frac{I_{w1}}{I_{w2}} \quad (2.16)$$

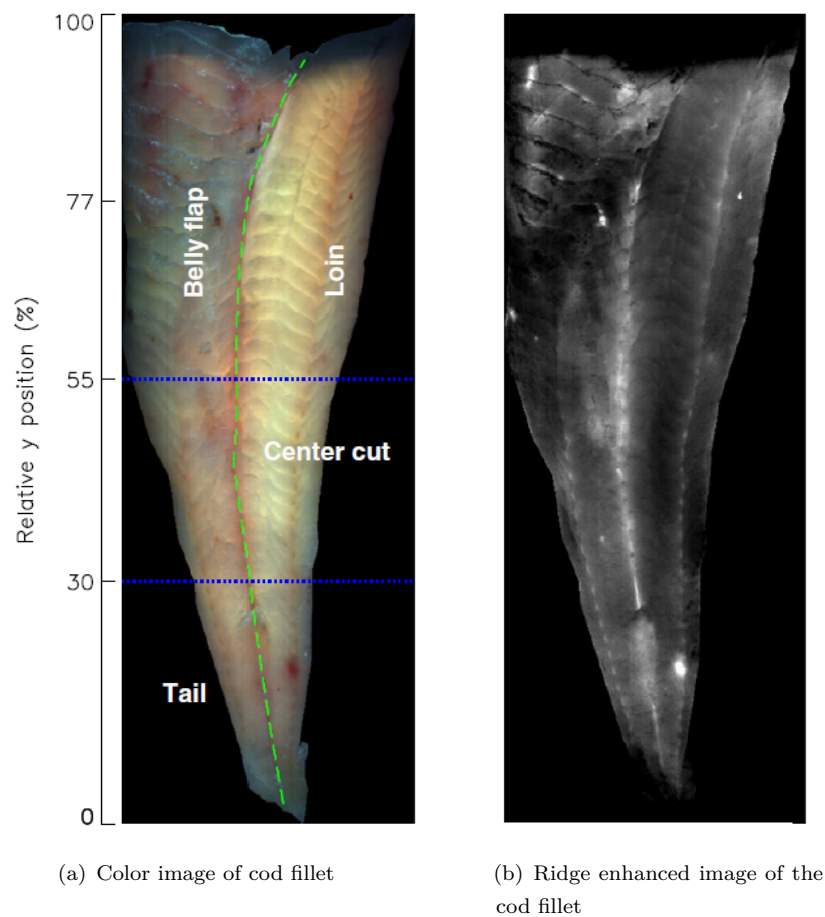
where  $I_r$  is the ridge enhanced image,  $I_{w1}$  is the image of wavelength band in which the ridge has high absorbance and the muscle has low absorbance and  $I_{w2}$  is the image of wavelength band in which the ridge has low absorbance than the surrounding muscles.

This is done so that the ridge image will have high values on the ridge and low values on the muscles.

Using the absorbance characteristics of haemoglobin this can be done by dividing the image band at 715 nm, which is a wavelength where absorption of hemoglobin is low and with the image band at 525 nm, a wavelength where absorption of haemoglobin is relatively high for all oxidation states.

Figure 2.5 (a) shows the colour image where the red, green and blue channels are represented by 640, 550 and 460 nm, respectively. The green dashed line indicates the manually detected centreline and the blue dotted lines indicate the transition between tail to centre cut and centre cut to loin/belly-flap respectively. Figure 2.5 (b) shows the centreline enhanced image. The axis on the left hand side indicates the position along the fillet in percent relative to fillet length.

This approach has been used to detect the center line of the cod fillet, where the blood concentration is relatively high. The process of ridge or centreline detection is crucial for determining the position of the blood spots on the fillet and has many applications in the quality assessment of fish. This ridge detection method can be thought of as a general method in the sense that it can be applied to other problems to find or follow ridges or valleys in hyperspectral images.



**Figure 2.5:** Colour image of cod (a) where green dashed line is the manually marked centreline, blue dotted lines indicate the transition between different parts of the cod and the corresponding ridge enhanced image (b). [16]

However, there are other effects that can yield a similar ratio of absorption between the two wavelengths, and the method is therefore not very robust. The method also does not utilize the full extent of information in the hyperspectral image.

## 2.5 Regression Models

The extensive data extracted from a HSI can be of many types, essentially either abundance data or raw spectral data, either of which can be modelled as a predictor variable with the corresponding chemical analysis results as the predictand using a regressor model.

Regression is a process of finding mathematical relationships among the variables. In a regression analysis, we consider a variable of interest which is the predictand and a number of factors that influence this predictand called the predictors, and try to

establish a relationship among them. Mathematically, we find a function that maps some independent feature variables to the dependent target variable.

The various supervised machine learning regressor models used in this comparative analysis study are discussed in detail in this section.

### 2.5.1 Linear model

A linear machine learning model essentially generates a linear combination of the variables to create a best-fit line to predict unknown values. It can also be thought of as line fitting models which might not be as predictive as a complex algorithm but can be trained relatively quickly, contain less number of parameters and are more straightforward to interpret. This section explains in detail about the linear models used in this thesis.

#### Linear Regression

Linear regression also called as Ordinary Least Squares (OLS) linear regression, fits a linear model with regression coefficients to minimize the residual sum of squares between the original observed targets, and the targets predicted by the linear approximation model.

Assuming a set of independent variables  $x = (x_1, x_2, \dots, x_r)$  where  $r$  is the number of predictors, linear regression assumes a linear relationship between the target variable  $y$  and  $x$  as:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_r x_r + \epsilon \quad (2.17)$$

where  $\beta_0, \beta_1, \beta_2, \dots, \beta_r$  are the regression coefficients and  $\epsilon$  is the random error. Now, the estimators of the regression coefficients also called predicted weights, are calculated as  $b_0, b_1, b_2, \dots, b_r$  which are used to define the estimated regression function

$$f(x) = b_0 + b_1 x_1 + b_2 x_2 + \dots + b_r x_r \quad (2.18)$$

The estimated or predicted response denoted by  $f(x_i)$  should be as close as possible to the corresponding actual response  $y_i$ . The differences  $y_i - f(x_i)$  are called the residuals.

A linear regression finds the best predicted weights  $b_0, b_1, b_2, \dots, b_r$  such that these correspond to the smallest possible residuals. To do this, we usually minimize the sum of squared residuals for all the observations. Hence, this method is also called ordinary least squares.

## Ridge Regression

As seen with Ordinary Least Squares Linear Regression maps a linear function to solve the relationship between input and output variables. The residuals  $y_i - f(x_i)$  in a ridge regression can be thought of as a loss function represented by:

$$\text{loss} = \sum_{i=0}^n (y_i - f(x_i))^2 \quad (2.19)$$

Mathematically, a single input-single output variable function maps out to form a line, whereas two input variables give a quadratic function or a function for a plane and so on. With increasing number of variables, the dimensionality increases as well and the coefficients of the model are found by minimizing the loss function in eq. 2.19.

One major issue with this method arises with increasingly large value of the coefficients leading to a possibly unstable model. This can be addressed by adapting the loss function to include additional costs for models with large coefficients like a penalty and such models are called penalized linear regression.

One way to penalize a model is based on the sum of the squared coefficient values  $\beta$ . This is called L2 penalty.

$$L2_{penalty} = \sum_{j=0}^p \beta_j^2 \quad (2.20)$$

An L2 penalty minimizes the size of all coefficients, and prevents any coefficients from being removed from the model by allowing their value to become zero. This penalty can be added to the cost function for linear regression and is referred to as Tikhonov regularization.

A special case of Tikhonov regularization is known as Ridge Regression which is especially useful to overcome the problem of multi-collinearity in linear regression. Multi-collinearity is the presence of linear relationships between independent variables leading to inaccurate estimates of the regression coefficients and thus hindering the model's ability to predict properly. This is commonly seen in higher dimensional models. By adding a degree of bias to the regression estimates, ridge regression reduces the standard errors. It can be written as:

$$\text{ridge}_{loss} = \text{loss} + (\lambda) * l2_{penalty} \quad (2.21)$$

Lagrange multiplier also called  $\lambda$  controls the weighting of the penalty to the loss function. A default value of 1.0 will fully weight the penalty whereas a value of 0 excludes the

penalty and reduces it to an ordinary least squares linear regression. Usually, very small values like  $10^{-3}$  or smaller are used for  $\lambda$ .

### Partial Least Squares

Partial Least Squares (PLS) can be thought of as a linear regression model that can be presented as a kind of simultaneous Principal Component Analysis (PCA) and regression. Partial least squares regression is a technique that reduces the predictors  $x$  to a smaller set of uncorrelated components and performs least squares regression on these components, instead of on the original data. It is also similar to Principal Component Regression (PCR), where the samples are first projected into a lower-dimensional subspace, and the targets  $y$  are predicted using transformed  $x$ .

Let  $X$  be the predictor matrix of dimensions  $(n \times N)$  and  $Y$  be the target matrix of dimensions  $(n \times M)$ , PLS transforms them from their original vector space into:

$$X = TP^t + E \quad (2.22)$$

$$Y = UQ^t + F \quad (2.23)$$

where  $T$  and  $U$  are the  $(n \times p)$  matrices of the latent vectors or projections of  $X$  and  $Y$  respectively,  $P$  is the  $(N \times p)$  matrix and  $Q$  is the  $(M \times p)$  matrix that represent the loadings or orthogonal weight matrices and are transposed, whereas  $E$  of dimension  $(n \times N)$  and  $F$  of dimension  $(n \times M)$  are residual matrices. The decompositions of  $X$  and  $Y$  are made so as to maximise the covariance between  $T$  and  $U$ .

PLS now constructs estimates of the linear regression between  $X$  and  $Y$  as:

$$Y = X\tilde{B} + \tilde{B}_0 \quad (2.24)$$

where  $\tilde{B}$  is the regression coefficient and  $\tilde{B}_0$  is the random error. PLS will iteratively estimate the values for  $\tilde{B}$  and  $\tilde{B}_0$  which will finally yield the least squares regression estimates  $B$  for  $B_0$  which are the corresponding values in the original vector space.

### 2.5.2 Non-parametric model

Non-parametric models are a set of non-linear machine learning models that do not make any strong assumptions about the data or the mapping function. They are less restricted and can learn any functional form from training the data and works without much prior knowledge about the data. These methods seek to best fit the training data in constructing the mapping function, while also maintaining the ability to generalize to

previously unseen data. This section explains in detail about the non-parametric models used in this thesis

### **K Nearest Neighbors**

One common example of the non-parametric models is the K-Nearest Neighbors (KNN) algorithm which predicts values based on how similar the new data instance is to the training patterns. This method does not assume anything about the data or the mapping function. It only assumes that the training patterns that are the most similar are most likely to have a similar result. This method stores all available cases from the training data and predicts the numerical target based on a similarity measure.

Assuming pairs of input data  $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$  and we need to predict  $y_r$  for a new data instance  $x_r$ , KNN algorithm first starts quantifying the input data points based on a similarity measure, usually a distance metric like Euclidean distance, Manhattan distance or Minkowski distance. Let us consider Euclidean metric for simplicity,

$$D(p, q) = \sqrt{\sum_{i=1}^n (q_i - p_i)^2} \quad (2.25)$$

where  $p$  and  $q$  are two points in space represented by  $(x_p, y_p)$  and  $(x_q, y_q)$ ,  $n$  is the number of dimensions of  $p$  and  $q$ .

It implements learning based on the  $k$  nearest neighbors of each data point, where  $k$  is an integer value specified by the user, usually chosen to be the square root of the total number of data points. For a given value of  $k$ , the algorithm will find the  $k$ -nearest neighbors of the given data point. It will then assign a class to the data point by choosing the class which has the highest number of data points out of all classes of the  $k$  neighbors. The  $y_r$  for our new data instance  $x_r$  is then computed by taking the average  $y$  value of the input's  $k$  neighbors. This algorithm is simple, but very successful for most regression and classification techniques.

### **Support Vector Machines**

Support vector machines (SVM) are a set of supervised, non-parametric learning methods used for classification, regression and outliers detection by finding a hyperplane in an  $N$ -dimensional space that can help distinctly classify the data points. The input data points are usually divided into two classes based on distance by a plane, the points that lie closest to both the classes are known as support vectors. The proximity between the dividing plane and the support vectors are calculated as the margin. SVM algorithm

now aims to maximize this margin and the optimal hyperplane is one which has the maximum margin.

The support vector regression (SVR) model is adapted from the SVM and does not depend on the distribution of the underlying independent and dependent variables. In contrast, SVR relies on kernel functions like linear, nonlinear, polynomial, radial basis function (RBF) or sigmoid. The basic idea behind SVR is that as long as the error is less than a certain value, there is no need to worry about prediction which is known as the principle of maximal margin. SVR can also be penalized using cost parameters, which is convenient to avoid overfitting.

SVR is a useful technique that provides users with a high degree of flexibility regarding the distribution of basic variables, the relationship between independent variables and dependent variables, and the control of penalty items. The model produced by Support Vector Regression depends only on a subset of the training data, because the cost function ignores samples whose prediction is close to their target.

### 2.5.3 Ensemble model

Ensemble methods is a technique that combines multiple simpler models to improve the predictive capacity of the system of models. Ensemble learning is a non-linear machine learning technique that helps improve machine learning results by combining several models as it allows the production of better predictive performance compared to a single model. Ensemble methods can be thought of as meta-algorithms that combine several machine learning techniques into one predictive model in order to decrease the variance using bagging methods, bias using boosting methods, or improve predictions using stacking methods depending on a chosen feature.

Two basic families of ensemble methods are usually distinguished and are averaging methods and boosting methods. The averaging or **bagging methods** build several estimators independently and then average their predictions producing a better model because its variance is reduced, like the Random Forests method. Whereas, **boosting methods** build the different estimators sequentially and tries to reduce the bias of the combined estimator which produces a powerful ensemble from several weak models, like Gradient Boosting method.

In order to understand Random Forests and Gradient Boosting, a basic understanding of Decision Trees is necessary. Decision trees are supervised machine learning models that determine the predictive value based on a series of questions and conditions. These models tend to be sensitive to the specificity of the data on which they are trained



meaning. If the training data is changed, the resulting decision tree will be most likely different and in turn the predictions will be different too.

### **Random Forest**

Random forests, also called random decision forests are a common ensemble method made up of individual decisions trees. They constructs many decision trees in parallel using a method called bagging. It outputs the mode of the classes of the individual trees for classification tasks or their mean prediction for regression tasks.

As discussed above, decision trees are computationally expensive to train, carry a huge risk of overfitting, and tend to find local optima because they cannot go back to the parent node after they have made a split. These weaknesses are addressed in a Random Forest which combines the predictive power of many decision tress and decreasing the variance.

Random forest is a bagging technique which means each tree in the ensemble is built from a sample drawn with replacement from the training set. These trees run in parallel and has no interaction among one another while training. The number of features that can be split on at each node is limited to some percentage of the total and is known as the hyperparameter. This ensures that the ensemble model does not rely too heavily on any individual feature, and makes fair use of all potentially predictive features.

Each tree draws a random sample from the original data set when generating its splits, adding a further element of randomness that prevents overfitting. These modifications help prevent the trees from being too highly correlated. Random forests achieve a reduced variance by combining diverse trees, sometimes at the cost of a slight increase in bias. In practice the variance reduction is often significant hence yielding an overall better model.

### **Gradient Boosting**

Gradient Boosting (GB), also called Gradient Tree Boosting or Gradient Boosted Decision Trees is an ensemble method that relies on boosting which is decreasing the bias of the system. Boosting can be thought of as a method of converting weak learners into strong learners. GB builds an additive model in a forward stage-wise fashion, sequentially combining many decision trees and can be applied to both classification and regression tasks.

As discussed above, the weaknesses of decision trees are addressed by the gradient boosting algorithm by using gradients in the loss function. The loss function is a measure

indicating how good the model's coefficients are at fitting the underlying data. In each stage, a regression tree is fit on the negative gradient of the given loss function. Gradient boosting allows the user to optimise the cost function based on the data at hand.

In gradient boosting, each new tree is a fit on a modified version of the original data set one on top of the other. It begins by training a decision tree in which each observation is assigned an equal weight, and after evaluation of the first tree, the weights are adjusted in such a way that the difficult to classify nodes are given relatively more weight than the easily classifiable ones. Thus, the second tree is built on these adjusted weights and so on. This improves upon the predictions of the previous tree. Subsequent trees get better at prediction than its predecessor and the final ensemble prediction is calculated as the weighted sum of the previous predictions.

#### 2.5.4 Neural Network model

Neural networks also known as Artificial Neural Network (ANN), are a set of non-linear machine learning algorithms that are developed to resemble the human brain to enable pattern recognition which can be further used in classification and regression tasks and reinforcement learning tasks. Neural networks are known to produce accurate approximations to any functions and have the ability to quickly learn and generalize problems which makes them versatile to use. The basic computing unit of a neural network is called a Neuron or **perceptron** and a group of perceptrons together form a layer. Many such layers are stacked together in numerous ways to parallelly process the input data.

##### Multi-layer Perceptron

Multi-layer perceptron (MLP) is a class of deep artificial neural network (ANN) and as the name suggests, is composed of multiple layers of perceptrons. The predictive capability of neural networks can be attributed to their hierarchical or multi-layered structure. This enables them to automatically capture the nuance features at different scales or resolutions and combine them into higher order features.

MLP makes no assumptions about the distribution of the data, the linearity of the output function or the predictor variable, or the type (measure) of the output variable. MLP consists of multiple parallel layers of nodes connected by weighted links. An MLP consists of an input layer, an output layer and a number of hidden layers in between them. Each perceptron or neuron has an activation function like tanh or hyperbolic tangent,

sigmoid, logistic and softplus which performs complex mathematical manipulations on the input data to provide a corresponding output.

A generic MLP train on a set of input-output pairs and learn to model the correlation or dependencies between them. It also involves adjusting the parameters, or the weights and biases, of the model in order to minimize error. During the process of training, MLP utilizes a supervised learning technique called backpropagation which is used to make those weights and bias adjustments relative to the error so as to obtain a minimum error.

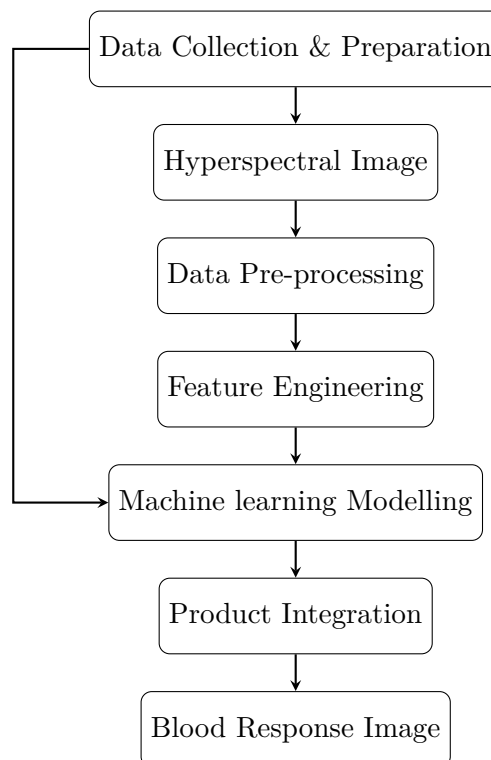


# Chapter 3

## Methodology

This chapter explains the methods and steps involved in detecting the presence of blood and determining its concentration in Salmon fillets. A basic outline of the methods involved are discussed in section 3.1 and a detailed explanation of the stages involved in this thesis are elaborated in section 3.2.

### 3.1 Outline



**Figure 3.1:** Outline of the thesis

This thesis focuses on a novel approach to detecting the presence of blood in a salmon fillet and involves complex data pre-processing techniques, experimentation and modelling. Figure 3.1 represents a basic overall outline of the steps involved.

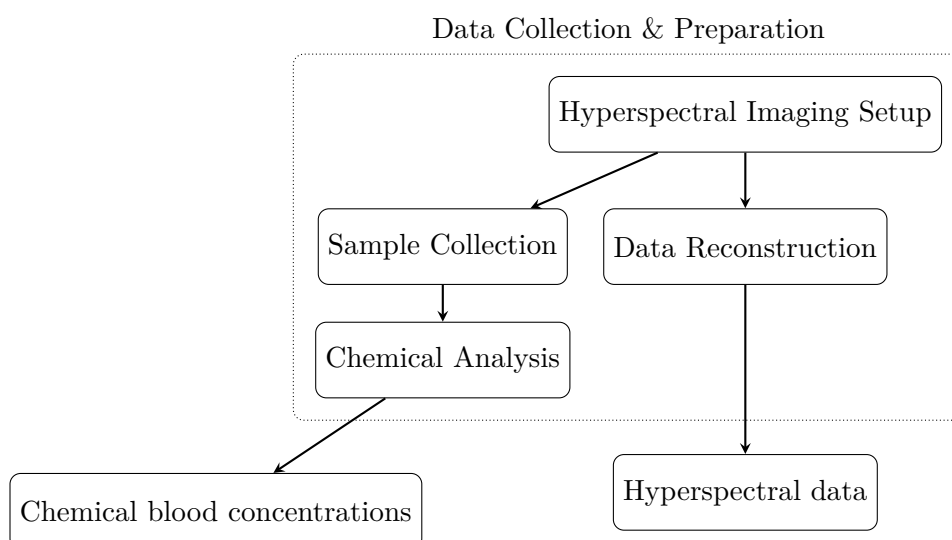
This thesis explores the multiple phases of data collection and preparation followed by the different data pre-processing techniques using hyperspectral image processing algorithms which are then used to extract the features in a feature engineering step. The machine learning models whose theoretical background was discussed in section 2.5 are then applied on the extracted features to predict the presence of blood and its concentration in the fillet pixels. These models are further evaluated and are integrated with the Maritech Eye product as presented in section 1.3. The final output will be a blood response image which will be depicting the varying concentration of blood on different parts of the Salmon fillet.

## 3.2 Proposed Solution

As presented in the section 3.1, an outline of the many steps involved in this study are represented in 3.1 which can be further broken down into major phases. An expanded view into each of these major phases can help understand the work done in this thesis better and the following sections are dedicated to this purpose.

### 3.2.1 Data Collection & Preparation

Consider an expanded view of the data collection and preparation stage in figure 3.2.



**Figure 3.2:** Expanded view of the steps of Data Collection & Preparation

The data collection starts with an introduction about the source of the fish fillets used in this study which will be scanned under the hyperspectral camera whose setup is described in the hyperspectral imaging setup section. This is then followed by sample collection and data reconstruction. Sample collection is the process by which the fillet muscles from the salmon at specific locations are collected for subsequent chemical analysis. Data reconstruction involves the assembly of the raw binary data from a hyperspectral camera to a more readable and manipulable format for processing.

The results of the data reconstruction stage is what we call the hyperspectral data which will be used in the pre-processing stage, while the results of the chemical analysis gives the haemoglobin absorbance values which will be used in the modelling stage as the predictand variable.

Each of the sub-stages explored in the data collection and preparation phase is further explained in detail in a dedicated section 4.1 in chapter 4.

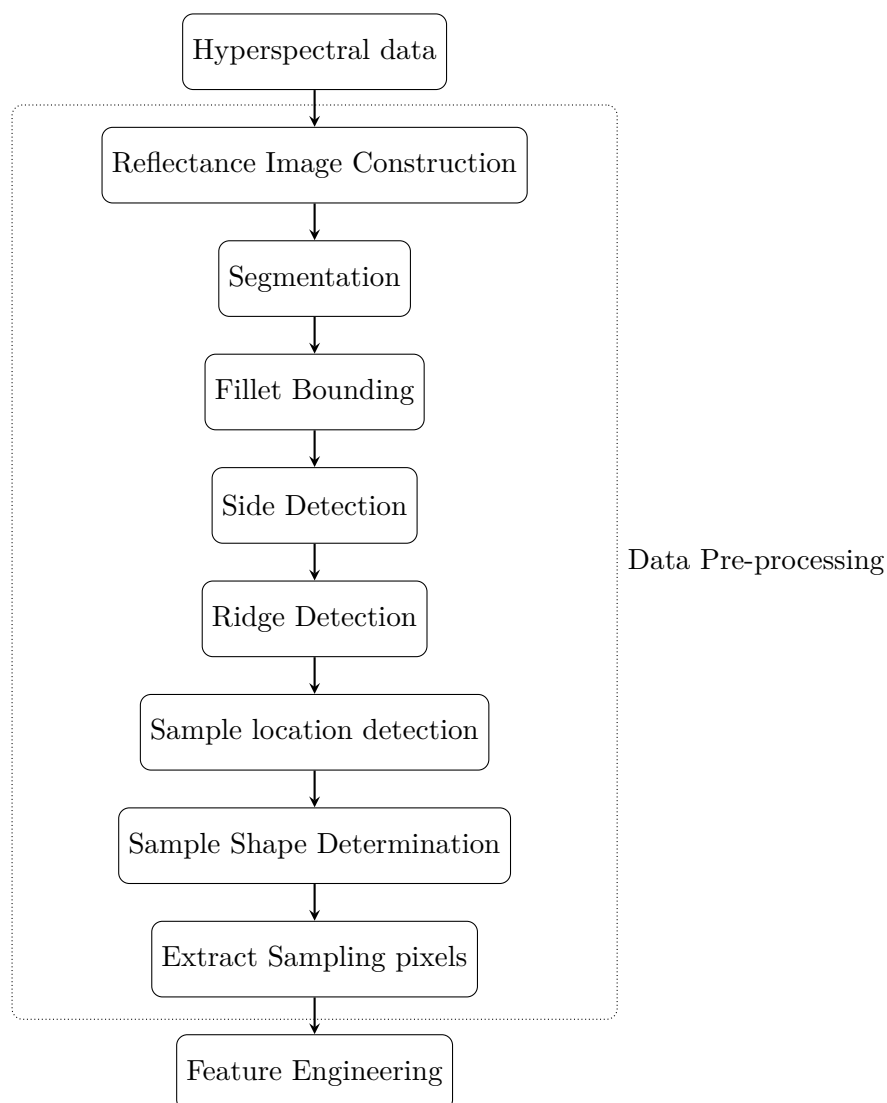
### **3.2.2 Data Pre-processing**

Consider an expanded view of the data pre-processing stage in figure 3.3. The data pre-processing stage consists of multiple sub-stages starting with the reflectance image construction. It is the process of using the dark and white references to construct the reflectance image on which all further processing will be done. This is followed by segmentation where the fish fillet pixels are separated out from the background pixels.

Segmentation is then followed by the fillet bounding step which ensures the fillet is straight and rotates it otherwise and that all the fillet pixels are bounded inside a region of interest. The next step involves automatic determination if the chosen fillet is left or right side of the salmon and is called the side detection.

This side detection step is followed by a very important step which has many other applications in the seafood industry, the ridge detection and enhancement. This step revolves around finding the centreline or spine of the salmon which is crucial in understanding the distribution of blood in a fillet. The muscles around the spine are rich in arteries and veins. The residual blood is due to improper bleeding of the fish. Once the fish is euthanised, the gills are cut to drain the blood from the largest vessels. This has to be done quickly and correctly to avoid residual blood in the fillet.

The main purpose of all the above preprocessing steps is to prepare the ground for finding the exact locations from which sample muscles were collected for the chemical analysis. The next step is to find those sample locations and is followed by the sample shape determination step which is further followed by the extraction of sampling pixels. The



**Figure 3.3:** Expanded view of the steps of Data Pre-processing

pixels from this sampling region also called the sampling pixels are used to extract the features necessary for the modelling process. These features are analysed and modelled with the chemical analysis results of the corresponding sampling sites.

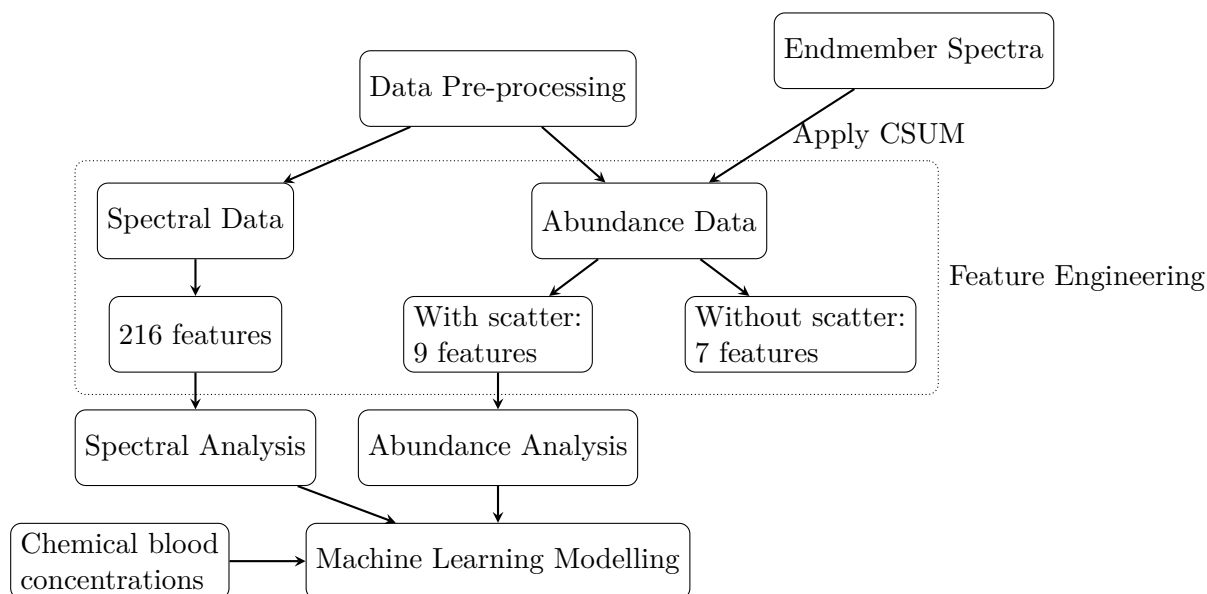
The importance of each of these individual steps are discussed explicitly in detail in a dedicated section 4.2 in chapter 4.

### 3.2.3 Feature Engineering

Consider an expanded view of the feature engineering stage in figure 3.4. This phase mainly focuses on the extracted sampling pixels from the pre-processing stage to find features that can contribute to the modelling process. The extracted pixels are used to get two different types of data, namely **spectral data** and **abundance data**, both



of which will be used to make two different frameworks of models. The spectral data is the direct raw pixels from the sampling sites which have 216 wavelength bands and thus 216 features. The abundance data is obtained by applying the Constrained Spectral UnMixing (CSUM) algorithm on the extracted sampling pixels. This will further be of two types, with and without scattering, of which only the data with scattering is considered for modelling. As can be seen in the figure 3.4, these two types of extracted features will be subject to normalization procedures before being used as the predictor variables for their respective modelling processes.

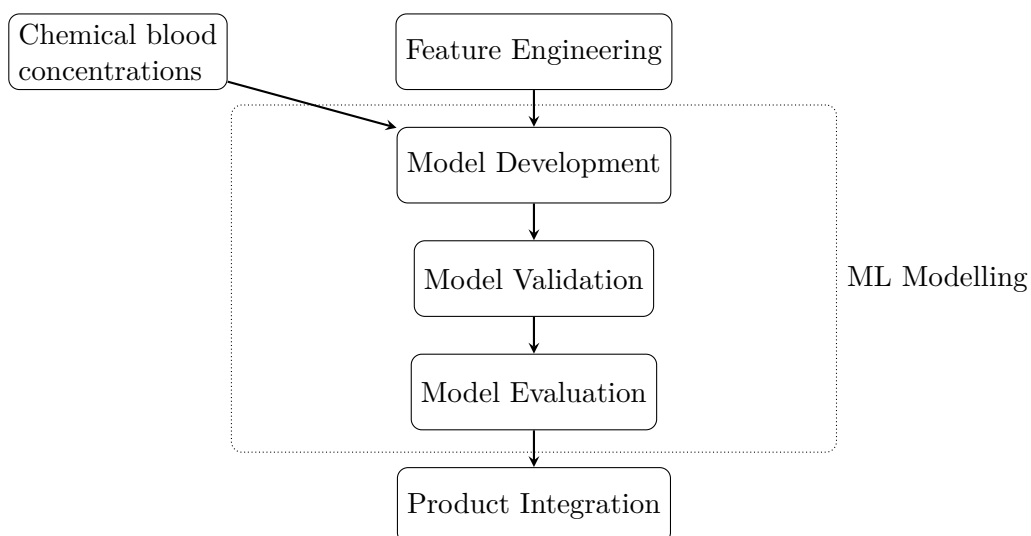


**Figure 3.4:** Expanded view of the steps of Feature Engineering

This phase is discussed more in detail in a dedicated section 4.3 in chapter 4.

### 3.2.4 Machine Learning Modelling

Consider an expanded view of the Machine Learning (ML) modelling stage in figure 3.5. The modelling stage can be divided into three major phases namely, development, evaluation and selection. The extracted features from the feature engineering stage as presented in section 3.2.3 is of two types and hence will have two frameworks, the spectral analysis and abundance analysis. These features along with the haemoglobin values from the chemical analysis in the data collection and preparation step discussed in section 3.2.1 will be used to build regression models. All the eight machine learning models discussed in section 2.5 have been implemented with both the spectral and abundance data respectively for the regression analysis and the results thus obtained are validated and evaluated using cross validation and regression evaluation metrics respectively. The best model in terms of performance and minimum errors is then selected for both data types and will then proceed to be integrated with the product.



**Figure 3.5:** Expanded view of the steps of Machine Learning (ML) Modelling

The different model parameters and evaluation metrics employed in this study are further discussed in detail in section 5.2, section 5.3 and section 5.4 of chapter 5.

### 3.2.5 Product Integration

The models thus selected after the evaluation phase are then converted to the Open Neural Network eXchange format (ONNX), which is an open format built to represent machine learning models in various softwares. ONNX defines a common set of operators and a common file format to enable ML developers to use models without dependency issues on different frameworks, tools, runtimes and compilers<sup>1</sup>.

The Maritech Eye product is integrated with the hyperspectral image analysis software called Breeze, developed by Prediktera<sup>2</sup> which is a complete suite of software to support workflow and development process of a hyperspectral imaging setup. The developed models will be integrated into this software and thus into the product itself. The process of product integration is further discussed in detail in 6.4.

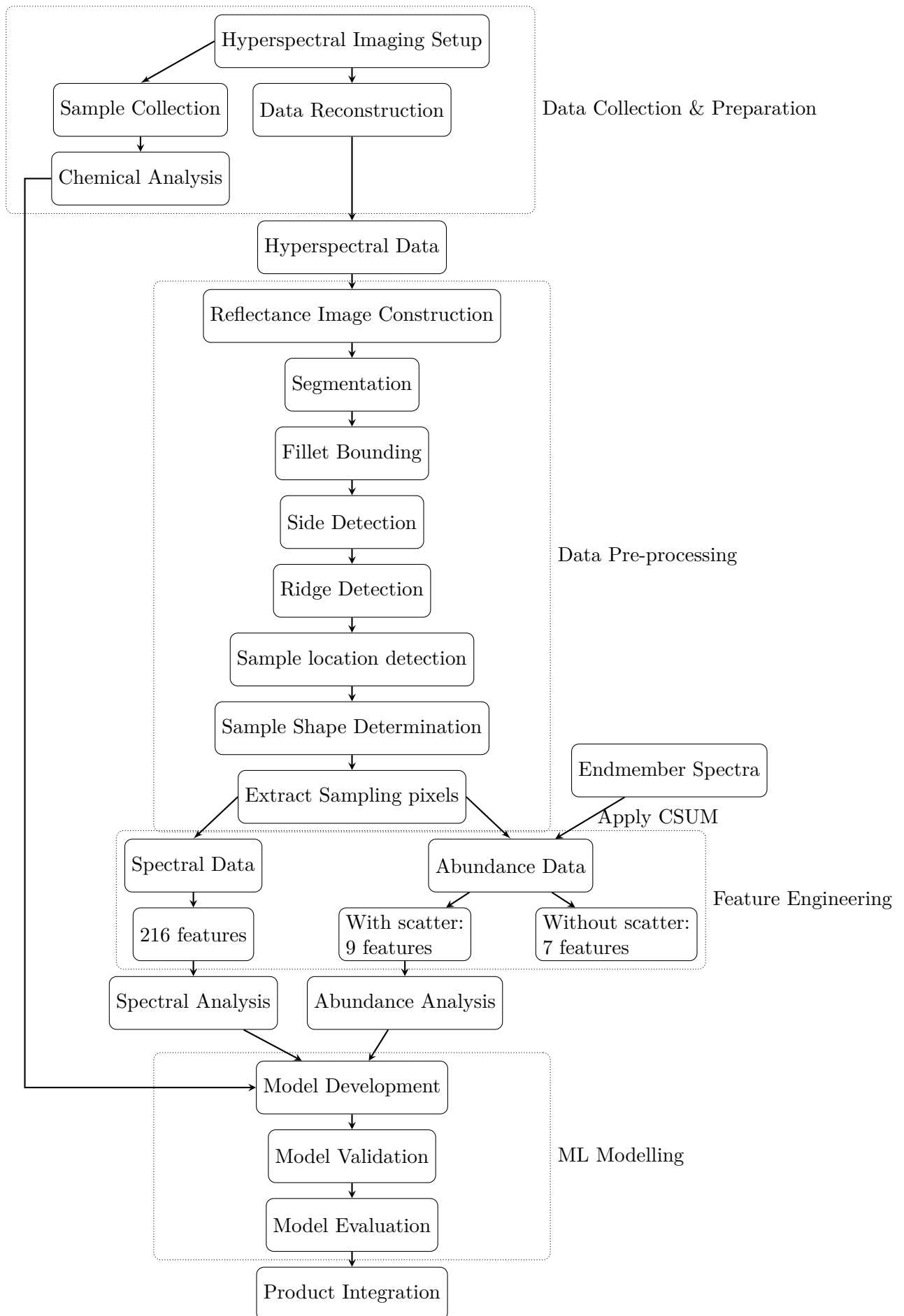
### 3.2.6 Summary

This novel approach to blood detection in salmon contains multiple steps which in themselves are of high industry value in terms of applications. All the phases involved in this thesis were explained one at a time in an elaborate way in the previous sections.

<sup>1</sup>For more details, refer to [ONNX](#)

<sup>2</sup>For more details about the software, refer: [Prediktera](#)

Figure 3.6 represents all these expanded phases combined. This figure depicts all the processes and steps involved in this thesis.



**Figure 3.6:** An overall expanded overview of all the steps involved in this thesis

## Chapter 4

# Research Data and Analysis

This chapter focuses on the research data and the subsequent pre-processing and analysis. The process of data collection and preparation are discussed in detail in section 4.1, the pre-processing procedures are explained in section 4.2, and the ensuing implementation of analyses and feature engineering is discussed in section 4.3.

### 4.1 Data Collection And Preparation

This section explains in detail the processes involved in the collection of hyperspectral data, the imaging setup, sample data extraction, chemical analysis and details about the endmember spectra.

#### 4.1.1 Introduction

The experiment was conducted on randomly selected 40 whole fishes of Atlantic Salmon farmed and harvested at Buksevika <sup>1</sup> by Mowi on 16 February, 2021; packed and stored on ice and transported to Nofima, Tromsø. The experiments were performed by Stein-Kato Lindberg and his crew including Stein Harris Olsen, the scientist who performed the chemical analysis, Torbjørn Tobiassen, scientist, Tatiana Ageeva, scientist, Margrethe Esaiassen, guest scientist and Iver Hovstad Raphaug, intern who all contributed to the filleting and manual sampling. I could not personally participate in the data collection process at Tromsø due to the COVID-19 restrictions at that time.

---

<sup>1</sup>Salmon farm at Buksevika, Norway [Google Maps Location](#)

The fishes used in this study were intentionally stored on one side during shipment in order to increase the blood concentration on one side and lower the concentration on the other side, thus providing us with fish fillets of high variance in blood content.

The samples were hand-filletted on 23 February, 2021 and the subsequent chemical analysis was done between 24 February, 2021 and 17 March, 2021. The pre-harvest analysis on the Atlantic Salmon fishes conducted on 1 February, 2021 were recorded and further details about the same can be found in the health certificate of the Atlantic Salmon batch attached in the appendix.

It is to be noted that this thesis should not be used to interpret the quality of the fish supplied by Mowi to the market. The salmon fillets used in this thesis are particularly prepared for this thesis alone.

This experiment begins with hyperspectral scanning of the collected fillets, followed by sample extraction and subsequent chemical analysis on the extracted samples. This procedure is explained in detail in the following subsections.

#### **4.1.2 Hyperspectral Imaging Setup**

The samples were first filleted<sup>2</sup> and scanned on a conveyor belt moving at a speed of 400 mm/s under the hyperspectral camera setup in the VIS-NIR wavelength region with a frame period of 1400 microseconds.

Figure 4.1 shows 20 salmon fillets tagged and ready to be scanned using the hyperspectral imaging setup.

The hyperspectral imaging setup consists of a pushbroom hyperspectral camera with a spectral range of 400 nm -1000 nm and the field of view is 0.28 mm across-track and 0.56 along track (Norsk Elektro Optikk, model VNIR-1024)<sup>3</sup> with a lens of focus 1000 mm was mounted 1020 mm above the conveyor belt. The samples were illuminated using two custom made fiber optic line lights (Fiberoptics Technology Inc. Connecticut, USA), fitted with custom made collimating lenses yielding light lines approximately 5 mm wide (Optec S.P.A., Milano, Italy). Each line light was 400 mm wide, with six bundles of optical fibers. The light from 12 focused 150 W halogen lamps with aluminium reflectors (International Light Technologies, Massachusetts, USA, model L1090) was fed into the fiber optic bundles, representing a total 1800 W of electrical input power. An illustration of the imaging and illumination setup is shown in Figure 4.2.

---

<sup>2</sup>Filletting is the process of removing the spine and cutting the fish length-wise. A Salmon can be cut into utmost 2 fillets (The left and right fillets).

<sup>3</sup>Product link: [HySpex VNIR-1024](#)

**Figure 4.1:** Salmon fillets tagged and ready to be scanned

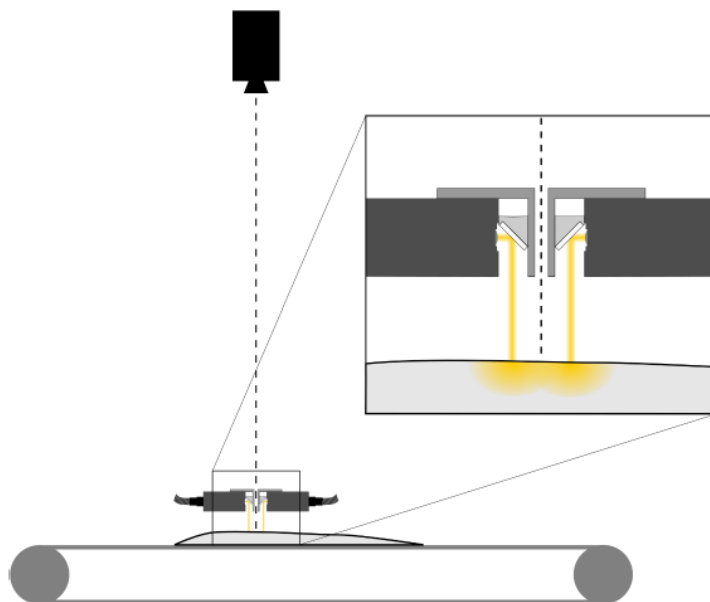
It can be seen that the light is reflected via mirrors and down on to the sample. The hyperspectral camera's line of view (dotted-line in figure 4.2) is placed directly between the two light lines. The transport of light from the light lines to the measurement area is due to scattering inside the sample.

Figure 4.3 shows the real-time hyperspectral imaging setup at the Nofima lab in Tromsø. The figure on the right shows one of the salmon fillets being scanned.

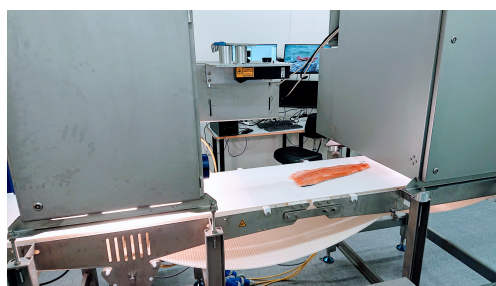
### 4.1.3 Data Reconstruction

The raw data captured by the hyperspectral camera is in binary format called Digital Numbers (DN format) and are stored in a .raw or .hispex file. A header file with the extension .hdr is also generated for every fillet that is scanned along with the raw file and contains the metadata. The crucial information from the header file like the dimension of the datacube is used to reconstruct the DN data from the corresponding raw image to a fully functional hyperspectral image. This is the image we use for further analysis.

**Figure 4.2:** An illustration of the Hyperspectral imaging setup [24]



**Figure 4.3:** Real-time Hyperspectral imaging setup at Nofima



#### 4.1.4 Sample collection

The samples required for the chemical analysis were cut out of the scanned fillets using a circular pipe of diameter 150 mm and weighing 10 g each in three spots at specific points relative to the centreline and the middle of the fillet. The muscles closest to the skin was removed to avoid the inclusion of any brown muscle.



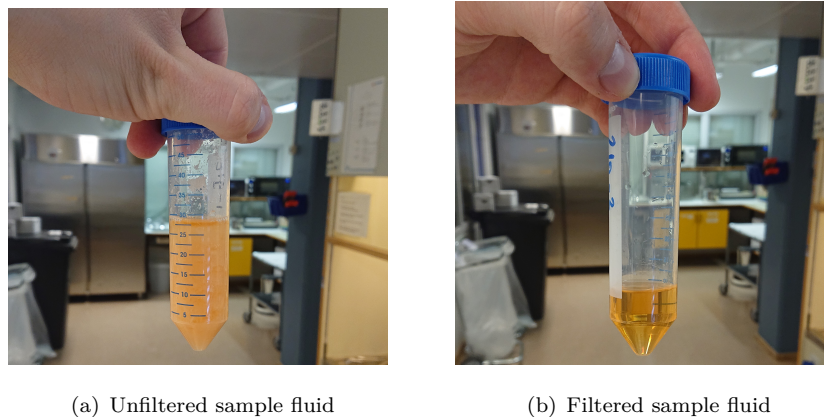
**Figure 4.4:** Sample collection by Nofima crew

Figure 4.4 shows the samples being cut out using the aforementioned circular pipe and then collected and freeze dried in the tubes and stored at  $-40\text{ }^{\circ}\text{C}$ .<sup>4</sup>

#### 4.1.5 Chemical Analysis

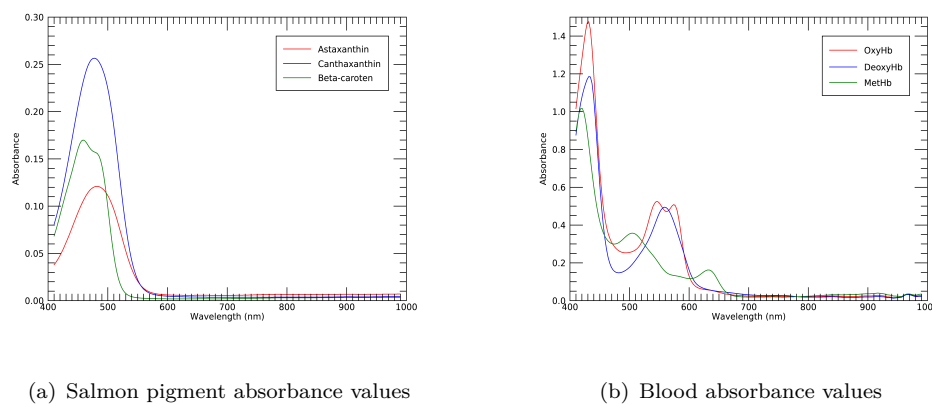
The chemical analysis starts with converting the haemoglobin to haematin, which is obtained by removing other proteins from the haem and oxidising the iron atom. This is done by coarsely grinding the frozen muscles and adding a mixture of 20 mL acetone, 4.5 mL water and 0.5 mL hydrochloric acid to it. These tubes are then shaken well and put in a fridge for an hour. The fluid from each sample tube was then filtered into another tube and the solid matter was discarded. Figure 4.5 shows the sample fluids before and after filtering [5].

<sup>4</sup>The photographs 4.1, 4.3, 4.4 and 4.5 were taken by Stein-Kato Lindberg on the premises of Nofima labs, Tromsø.



**Figure 4.5:** The sample fluids

The absorption of haemoglobin and the other pigments present in salmon have similar spectral signatures. Figure 4.6 shows the absorbance values of the salmon pigments and the different oxidation states of blood plotted on the Y-axis against the wavelength values on the X-axis.

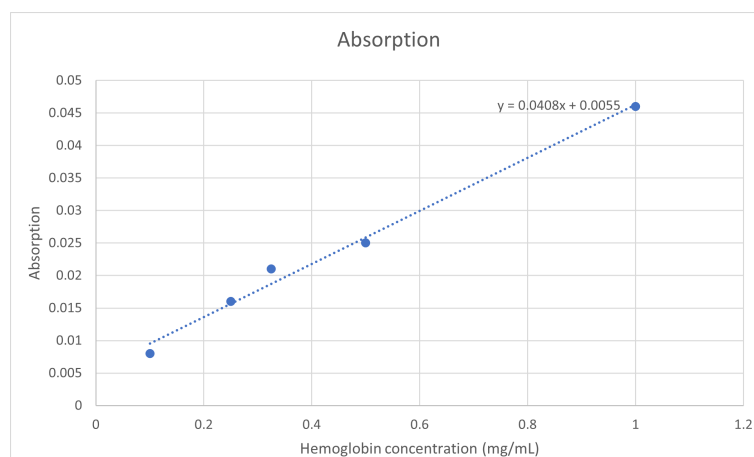


**Figure 4.6:** The absorbance values of the salmon pigments (a) and blood (b)

In Norwegian salmon the most common pigments in the feed are astaxanthin, betacaroten and cantaxanthin and are the pigments present in the salmon fillets. There are some other pigments that they use elsewhere but is not relevant to this work. The spectral signatures of these pigments are plotted in figure 4.6 (a). It can be seen that they have peak values in the range 450 nm to 550 nm approximately. The different oxidative states of blood are oxy-haemoglobin, deoxy-haemoglobin and met-haemoglobin whose spectral signatures are plotted in figure 4.6 (b). It can be seen that they have peak values in the range 400 nm to 480 nm and again smaller peaks between 510 nm and 600 nm.

The peak values of absorbance in the plots in figure 4.6 represent the wavelength regions best suited to identify the corresponding chemical substances. The overlap of spectral signatures of the pigments and blood absorbance values makes it difficult to detect only blood. Whereas haematin has an absorption peak at a higher wavelength than that of haemoglobin. Haematin is a compound derived from haemoglobin by removal of the proteins and oxidation of the iron atom. This conversion of haemoglobin to haematin is done to ensure that the pigments of the salmon do not interfere with the measurement of the blood spectrum.

**Figure 4.7:** Calibration curve



The filtered fluid recovered from the samples were analysed with a Shimadzu UV-1800 spectrophotometer operating at 640 nm. The absorbance from each sample was recorded and a calibration curve was made using bovine haemoglobin at different concentrations diluted with the same sour acetone mix as was used for the salmon samples. This accounts for the shift in the spectral values. Figure 4.7 shows the calibration curve plotted for 4.5 mL of the filtered fluid. Concentrations were calculated from the absorbance values by means of the calibration curve.

## 4.2 Data Pre-processing

The hyperspectral image data from scanning the 40 tagged fillets (numbered from 177 through 216) were of the format .raw and had an Environment for Visualizing Images (ENVI) header file that contains metadata and uses the same name as the image file, with the file extension .hdr. This contains the metadata like date and time of acquisition, description like frame period, number of frames, camera temperature, and aperture size.

Among them, the "wavelength" metadata represents all the 216 wavelength bands in which the data has been captured and the "default bands" metadata has the numbers of

the wavelength bands which represent the default RGB colour bands (here observed to be band numbers 55, 41, and 12) which can be used to display the hyperspectral data as an RGB image thus visible to the naked eye. Figure 4.8 shows the RGB representation of the hyperspectral image of the fillet with sample number 177.

**Figure 4.8:** RGB image of fillet sample 177



### Constructing Reflectance Image

There are in total 40 folders and each one containing the image data file which is named "measurement" in both .raw and .hdr formats along with a white reference and dark reference images. A 20 mm thick plate of polytetrafluoroethylene (PTFE, also known as Teflon) was used to create the white reference image prior to each imaging session. The physical characteristics of PTFE remain stable over time and it reflects well on all the wavelengths in the visible range. And thus images of the PTFE can be used to measure and correct for changes in illumination and camera sensitivity over time. The dark reference image is captured with no light input, essentially a black image. The image calibration is usually done to compute the reflectance image. The formula to compute the reflectance image is as follows:

$$I_{reflectance} = \frac{I_{measurement} - I_{darkref}}{\text{mean}(I_{whiteref})}$$

Here, the mean is taken over the along-track dimension, and this formula is applied line by line in the image.

The reflectance image is computed to locate the positions from where the sample muscles were collected for the chemical analysis. Calculating the spectral mean of the pixels

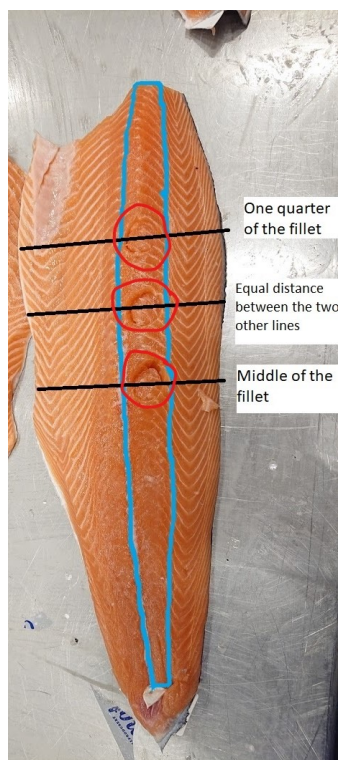
from these sample locations will give us the predictor variable  $x$  while the corresponding chemical analysis results will act as the predictand variable  $y$  for our spectral regression models.

### 4.2.1 Determining Sampling locations

The position of the sampling locations on each fillet needed to be calculated so as to link the results of the chemical analyses to the corresponding pixel locations on the hyperspectral images of the fillets. As was discussed earlier, one whole salmon can be cut into two fillets, right and left fillets.

Figure 4.9 shows the 3 spots from where the samples were collected from one of the fillets. First, from the absolute middle of the fillet, the other from one quarter of the neck (thicker end) of the fillet which is at the one-quarter point of the fillet and the last one from the middle of the two former obtained points. All these points were cut out from above the spinline of the fish to avoid major arteries and blood vessels. Now, these spots in the hyperspectral images can appear either to the right or left of the centreline depending on whether it is a right fillet or left fillet.<sup>5</sup> In figure 4.9, the samples appear to be obtained from the right side of the ridge since it is a right fillet.

**Figure 4.9:** Locations of Samples obtained for chemical analysis



<sup>5</sup>The sampling locations can be found on the left of the centreline for a left fillet and right of the centreline for a right fillet.

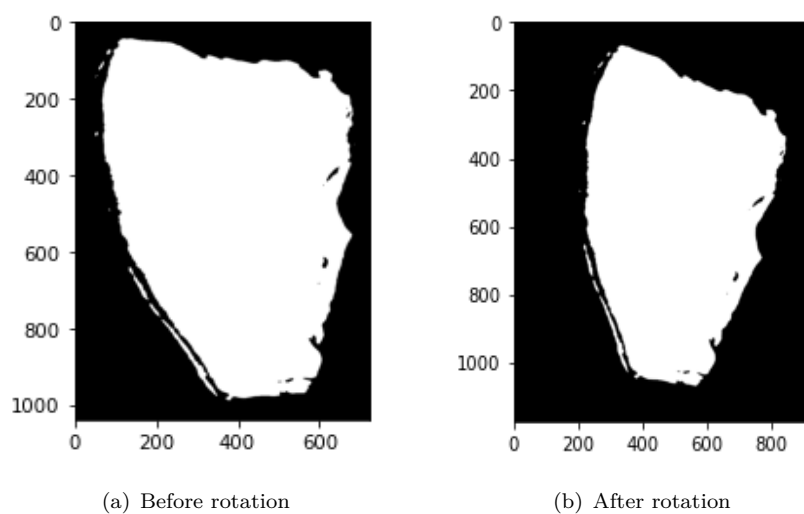
The steps involved in determining the sampling locations are explained in detail in this section below.

### Step 0: Segmentation

Before starting to work on the hyperspectral image, it is important to perform segmentation on it to come up with a mask that can be used for background exclusion, deciding the side of the fillet, and locating the sampling points. I have applied Gaussian blurring followed by Otsu thresholding to create a segmented mask in which the fillet pixels are displayed in white and the background pixels in black. Considering this, segmentation as an essential part of pre-processing.

### Step 1: Bounding the fillet

To find sampling locations, it is necessary to first rotate the hyperspectral image to make sure the centreline of the fillet is straight and aligns vertically with the y-axis of the image. The angle of rotation was found by applying PCA (Principal Component Analysis) to the image array and then rotating the image by that corresponding angle.

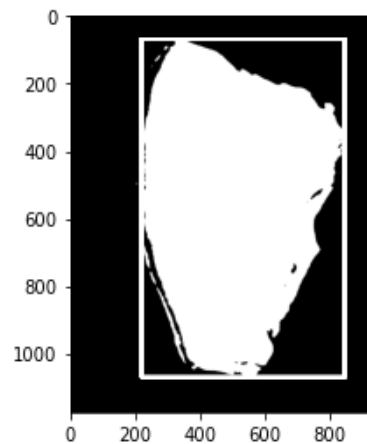


**Figure 4.10:** The segmented image of fillet 177 before and after rotation.

Figure 4.10 shows the segmented image of the hyperspectral image of fillet number 177 <sup>6</sup> before (a) and after (b) rotation. This rotated image is then used to find the mid point of the fillet. To do that, a bounding box needed to be drawn around the fillet which was done using the contour function from OpenCV. Figure 4.11 shows the corresponding bounded image.

---

<sup>6</sup>It is to be noted that all example images shown hereon is of the fillet number 177

**Figure 4.11:** Segmented and rotated image after bounding.

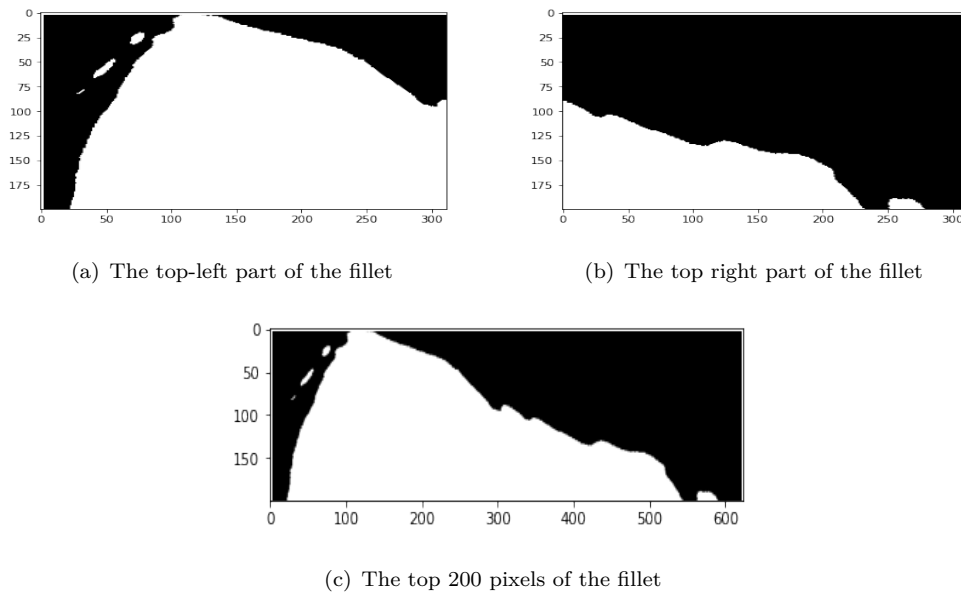
### Step 2: Detecting the side of the fillet

Now, it is important to figure out if the fillet is a left fillet or right fillet to be able to decide if the sampling locations are on the left or right of the centreline. The side of the fillet can be manually decided by a human by looking at the neck or the thicker end of the fillet. For example, a right fillet will have a head that slopes down to the left due to the filleting process where the head and the gills were removed.<sup>7</sup> Observing figure 4.9, it can be understood that it is a right fillet since the right upper corner seems thicker and slopes down towards the left. Whereas in figure 4.8, the left top corner seems thicker and it slopes towards the right making it a left fillet.

Automating this process can be done in several ways. The way I chose to do it is using segmentation. First a segmented mask from the hyperspectral image was obtained. This enabled in counting the number of white pixels in the upper left and right corners of the image to compare which side had more white pixels and thus determining which side of the image had more fillet pixels. The side with more number of white (or fillet) pixels meant it was longer on that side and thus helping in determining which sided fillet it was.

---

<sup>7</sup>For better understanding, please look at this [video](#) of salmon filleting.



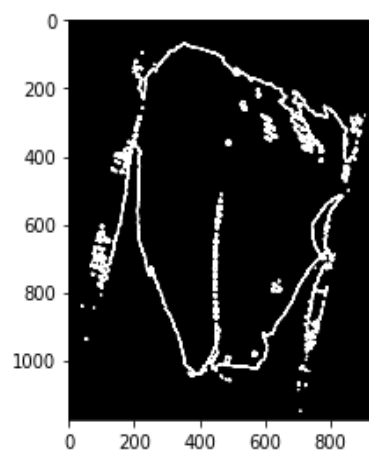
**Figure 4.12:** Applying side detection on fillet 177

Figure 4.12 (c) shows the segmented top 200 pixels of the fillet 177, the top left part (a) and the top right part (b). Counting the number of white pixels (fillet pixels) and comparing the two sides, the top-left image has more number of white pixels as can be seen. This shows that fillet 177 is thus a left fillet.

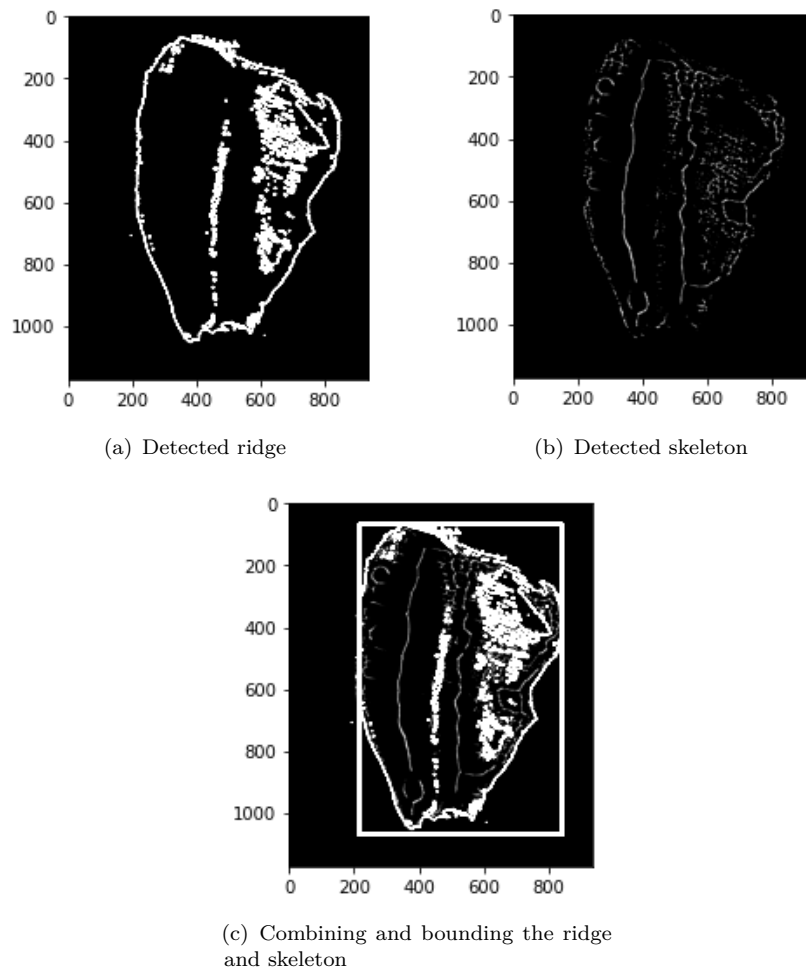
### Step 3: Ridge detection and enhancement

As discussed above, the samples were collected from above the centreline of the fish, which is either on the right or the left side of the centreline in the fillet. This step requires detecting the ridge in the hyperspectral image as it is not visible in the segmented, bounded image in figure 4.11. The ridge appears in the enhanced image in the form of higher values on the centreline of salmon.

**Figure 4.13:** Ridge detected image using Skjelvareid et al. method.







**Figure 4.14:** Applying step 3 on the image of fillet 177.

Ridge detection for cod fish has been done before by Sivertsen et.al, in [16] which is described in detail in section 2.4.

Applying this method of ridge detection to this problem, the two wavelength band values for this case were found to be between 490-510 nm and 565-575 nm. The computed ridge detected image using this method is shown in figure 4.13.

This method, despite being successful with cod was not successful with salmon. As it can be seen from figure 4.13, the ridge disappears along the top and thus making it hard to locate the sampling points. Various morphological functions from OpenCV like erosion and dilation in different combinations were applied to this problem to highlight the ridge on the hyperspectral image but without success.

The method that worked best here is a combination of considering the reflectance image in YUV<sup>8</sup> color space and thresholding it. Followed by applying skeletonize function on

<sup>8</sup>YUV stands for one luma component (Y) and two chrominance components, called U (blue projection) and V (red projection)

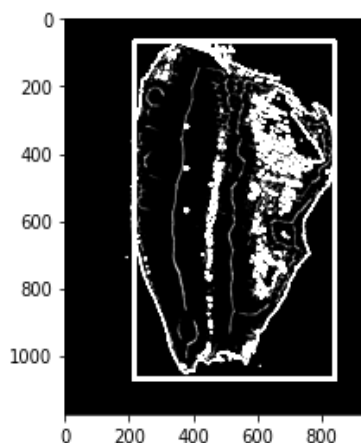
it to highlight the ridge and possible skeletal features. Figure 4.14 (a) shows the detected ridge and (b) shows the detected skeletal features while (c) shows the combination of the both along with the bounding box. It is important to notice a lot of false detections on the right end of the fillet in 4.14 (a), this is due to the high fat content in the belly of this left fillet which has a higher reflectance than the surrounding muscles similar to but not the same as that of the spine. Considering the detected ridge in figure 4.14 (c), the coordinates can then be calculated.

#### **Step 4: Computing the co-ordinates of the sampling location**

Looking back on figure 4.9, the samples were taken at calculated positions with respect to the ridge in the horizontal axis and the centre of the fillet in the vertical axis. To be able to correlate the pixel information from the hyperspectral image to that of the corresponding chemical analysis, the sampling sites from the original fillet needed to be positioned on the hyperspectral images. To be able to find the sample sites, their exact centres needed to be computed.

The y co-ordinates of the three sampling locations can be found using the bounding box. The vertical mid point of the fillet is found to be the exact mid point of the bounding box. The quarter point of the fillet is determined by finding the mid point of the former found point and the top half of the bounding box. But locating the x co-ordinates was not straight forward. As this was not constant and changed relative to the position of the detected ridge. It is known that the sampling locations lie on the left side of the ridge for a left fillet and vice versa, but exactly how far the distance from the centerline needed to be determined. The samples were collected approximately one third distance from the ridge to the corner of the fillet. This information helped determine the x co-ordinate of the centres. The computed centre co-ordinates are saved. Figure 4.15 shows the computed centre points as white points on the left of the ridge since fillet number 177 is a left fillet.

**Figure 4.15:** Computer centres of the sampling locations.

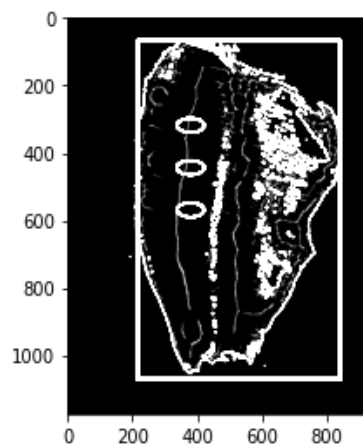


**Step 5: Determining the sampling shape**

Before proceeding further, it is necessary to take a look at the hyperspectral image and the normal photograph of the fillet. The aspect ratio is not 1:1 since the pixels are not quadratic, meaning each pixel in the hyperspectral image is of different dimensions than that of the actual fillet and hence the fillet in the hyperspectral image 4.8 looks shorter and wider than in the normal photograph 4.9.

The information about the pixel resolution of each hyperspectral camera is different and are stored in the header (.hdr) file of each image as “Pixelsize x” and “Pixelsize y”. These give the spatial resolution of each pixel per frame, i.e. the physical field of view of each CCD in the camera. The pixelsize x value (0.28 mm) can be used directly but since the fillet moves along the y direction during imaging, the speed of the conveyor belt and the frame period have to be taken into account. The true y resolution is the product of the conveyor belt speed and the frame period. The conveyor belt in this case moves with a speed of 400 mm/s and for the camera HySpex VNIR-1024, the frame period is set to 1400 microseconds, the product of which gives 0.56 mm, which is the same as the pixelsize y value in the header file of images captured by that camera. An important note is to remember that this is not the same with other cameras and the true y resolution needs to be calculated with caution.

**Figure 4.16:** Computed sampling locations.



After the true x and true y pixel dimensions are calculated, the sampling locations are marked on the segmented mask by scaling the axes of the ellipses to match the aspect ratio of the image, that is by scaling them from the original fillet dimensions (usually 1 mm x 1 mm) to that of the calculated hyperspectral image dimensions which turns the circles in fig 4.9 to ellipses in the hyperspectral image mask shown in fig 4.16 whose axes are calculated by scaling the radius of the circle ( $150 \text{ mm}/2 = 75 \text{ mm}$ ) using the

calculated pixel dimensions. The ellipse here has a minor and major axes of 21 and 42 respectively (0.28 x 75 mm and 0.56 x 75 mm).

This segmented mask and the computed co-ordinates along with the axes are used to capture the correct hyperspectral pixel information from the reflectance image of all the fillets.

### 4.3 Feature Engineering

The analysed hyperspectral data from section 4.2 results in the determination of sampling pixels. The pixels from these elliptical sampling sites were then averaged to make one spectrum for each sampling location, as the features which were then further processed to act as the predictor variables, while the haemoglobin count from the chemical analysis acts as the predictand variable for the regression models. The extracted pixels from the three sampling elliptical regions of each fillet for the 40 fillets would give 120 regions of pixels to be considered. Meaning there are 120 samples or rows to train the models on.

There are two types of features that were considered for modelling,

- Spectral data
- Abundance data

The calculation to obtain these two types of features and their subsequent processing are explained in detail in the following sections.

#### 4.3.1 Spectral data

Spectral data is the direct pixel information obtained from the sampling sites of each fillet. There are 216 wavelength bands that the hyperspectral camera captured of the fillet and hence there are 216 different values of data pertaining to each pixel of the sampling site. The pixels from each elliptical sampling sites are then averaged to give the spectral information from the 216 wavelength bands, meaning there are 216 features for every sampling location.

These 216 features were then subjected to **Standard Normal Variate (SNV)** which is a popular preprocessing method. SNV normalizes a spectrum by removing a constant offset term by subtracting the mean value of the full spectrum and brings all spectra to the same scale by subsequent division by the standard deviation of the full spectrum.

Mathematically, the standard normal variate of an instance  $x$  of a spectra can be represented as:

$$x' = \frac{x - \mu}{\sigma} \quad (4.1)$$

where  $x'$  is the normalized spectra,  $\mu$  is the mean and  $\sigma$  is the standard deviation of the value of the whole spectrum. After SNV, each spectrum will have a mean of 0 and a standard deviation of 1.

These normalized features will now act as the predictor variables of the spectral framework.

### 4.3.2 Abundance data

Abundance data is the data obtained from the abundance values calculated for each endmember by applying the CSUM on the pixels from the sampling sites. The endmembers for a salmon fillet as presented in section 2.3.3 are water, fat, muscle, pigments like astaxanthin, betacaroten and cantaxanthin, and blood (summing up the 3 oxidative states of haemoglobin), making up in total 7 individual endmembers. The CSUM algorithm is applied on the pixel sites with these 7 endmembers to find their respective abundance values giving 7 features for every sampling location.

One more important point to note about the CSUM algorithm explained in detail in section 2.4 is that it takes into consideration the effects of light scattering. To include the scattering effect, it is necessary to include two more features namely offset and ramp which attribute to the scattering effect. Thus giving 9 features in total for every sampling location.

These 9 features were then subject to rescaling using the **min-max normalization**. It is used to normalize the range of independent variables or features of data and is also known as data normalization.

Mathematically, the min-max normalization of an instance  $x$  of a spectrum can be represented as:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (4.2)$$

where  $x'$  is the normalized spectrum,  $\min(x)$  is the minimum and  $\max(x)$  is the maximum value of the whole spectrum. After min/max normalization, each spectrum will be rescaled to the range in between 0 and 1.

### 4.3.3 Outlier removal

Data preparation is an essential step before modelling and in this case, the most important task was outlier removal. An outlier, simply put, is any data instance that is either too big or too small to be true or in some way does not fit the pattern followed by the other data points in a dataset.

There are many potential sources of outliers in this study including but not limited to miscalculations during calibration in the chemical analysis. The process of muscle-fluid filtration explained in section 4.1.5 could be one such potential source. This filtration must result in a clear liquid for the spectrometer to measure the absorbance, but some samples do not always result in a clear liquid, which means that light scattering affects the absorbance calculation. And hence such samples can be categorised as outliers. Three such samples in this study were distorted due to scattering and thus needed to be removed before the modelling.

Another source of outliers is of course human error. Despite exercising utmost care and preparation, two such cases were detected. One case was contamination of the collected sample muscles with other particles in the lab while the other was mislabelling of the collected sample muscles. Four samples had to be categorised as outliers on this basis requiring 7 samples in total to be removed from the 120 samples.

## Chapter 5

# Model Development and Evaluation

This chapter represents the experimental setup and discusses in detail the model development, validation and evaluation. The technical details about the setup is elaborated in section 5.1, the model development, validation and evaluation are discussed in section 5.2, section 5.3 and section 5.4 respectively.

### 5.1 Technical Setup

The entire coding for this study has been done using Python 3.8.5 in Jupyter Lab and Spyder environments. Various libraries were used for this study such as:

- numPy (v1.19.5)
- pandas (v1.2.4)
- matplotlib (v3.4.1)
- spectralpython (v0.21)
- scikit-learn (v0.24.1)
- scikit-image (v0.17.2)
- opencv (v4.5.1.48)
- scientific-python (v1.6.2)
- pillow (v8.2.0)
- lxml (v4.6.3)

- imageio (v2.9.0)
- skl2onnx (v1.8.0).

The scikit-image library provides the functions for morphological functions and colour space conversions. The scikit-learn provides the functions for the machine learning models, spectralpython library provides most of the processing tools required to process a hyperspectral image while opencv aids with the basic image processing functionalities among others. The skl2onnx library is used to convert the scikit-learn models built into the ONNX format to allow product integration.

All the computations were done in a Lenovo ThinkPad configured with Intel Core i7 (9th generation) 9750HQ /2.6 GHz processor, 32GB DDR4 RAM and 16GB NVIDIA Quadro T1000 Graphics Card.

### **GitHub repository**

The GitHub repository containing the code developed in this thesis can be accessed here: [Github repo](#). This is a private repository, please contact me on my [mail](#) to gain access.

## **5.2 Model Development**

The feature engineering process as presented in section 4.3 gives two types of data. The spectral data with 216 features and the abundance data with 9 features. This results in turn in two types of model frameworks. Both frameworks will use the results from the chemical analysis discussed in section 4.1, the haemoglobin content computed as mg per g of muscle. A comparative regression analysis using the selected models discussed in section 2.5 is done with both types of data, which shall be explained in detail in this section.

The two types of data were used to build two separate frameworks using the same 8 regression models from section 2.5, implemented using the scikit-learn functions of the corresponding names. Both frameworks have a similar setup. The parameter tuning for each model in a framework was done using the GridSearchCV function, also part of the scikit-learn library. The dataset for each framework, was shuffled and then split into 60% train and 40% test set, and each of the models were subject to the Leave One Out (LOO) cross-validation technique for parameter estimation.



### 5.2.1 Parameter Tuning

A grid search is applied to a set of different values to each parameter of a model to find the best combination of the values for every model. Grid search applies every value to every parameter for every model to find the combination with the least possible error. The list of parameters and their corresponding set of values for each of the machine learning algorithms used are listed and explained in detail in this section.

#### Linear model

The linear models used in this study are Linear regression, Ridge regression and Partial Least Squares (PLS) Regression. The Linear regression and PLS Regression are fairly simple models and do not have parameters that need to be tuned using a grid search while ridge regression is a more tunable model with parameters that can be estimated using a grid search.

The parameters that needed to be tuned in the ridge regression model are 'alpha' and 'solver'. The 'alpha' is the regularization strength and improves the conditioning of the problem and reduces the variance of the estimates. It must be a positive value, the default value used by scikit-learn library being 1. A list of values ranging from 0 to 30 were provided to the grid search function to find the best estimate. The 'solver' is used in the computational routines and the default value used by the library is 'auto' which chooses the solver based on the data. The other available solvers that were given to the grid search function were 'svd', 'cholesky', 'lsqr', 'sparse\_cg', 'sag', and 'saga' <sup>1</sup>.

#### Non-parametric model

The non-parametric models used in this study are K Nearest Neighbors (KNN) and Support Vector Machines (SVM). Both these algorithms are relatively more complex than the linear models and thus have a number of parameters that can be tuned in order to get a model with improved performance.

The parameters of KNN that required tuning were 'weights' and 'algorithm'. The 'weights' parameter determines the weight function used in the prediction and has three possible values: 'uniform' which means all points will be weighted equally, 'distance' which means the points were weighted by the inverse of their distance and a third user defined metric. The default value used by scikit-learn library is 'uniform'.

The regression version of the SVM model, called the SVR model has 5 tunable parameters that are of interest in this study, namely 'kernel', 'degree', 'gamma', 'tolerance', and 'C'. 'kernel' as the name suggests specifies the type of kernel to be used in the algorithm.

---

<sup>1</sup>For more details on the solvers and their function. Refer to: [Ridge](#)

The possible kernels that can be used are 'linear', 'poly', 'rbf', 'sigmoid', 'precomputed' and the default value used by scikit-learn library is 'rbf'. 'degree' is the degree of the polynomial kernel and is ignored if a different kernel is used and has a default value of 3. 'gamma' stands for the kernel coefficient and has possible values 'scale', 'auto' or a user defined value with a default value of 'scale'. 'tolerance' is the tolerance for stopping criterion and has a default value of 0.001. 'C' is the regularization parameter whose strength is inversely proportional to C and must strictly be a positive value. It has a default value of 1.

### **Ensemble model**

The ensemble models used in this study are Random Forests and Gradient Boosting. These are fairly complex models similar to the non-parametric models. As ensemble models are essentially a collection of simpler models, an array of different parameters can be tuned.

Among the many parameters in the Random Forest implementation on scikit-learn, the parameters of interest for this thesis are 'max\_features', 'n\_estimators', 'bootstrap' and 'criterion'. The 'max\_features' decides the number of features to consider when looking for the best split and can be any of the values among 'auto', 'sqrt', 'log2' or a user defined value. It has a default value of 'auto'. The 'n\_estimators' parameter as the name suggests determines the number of trees in the forest to build and has a default value of 100. The 'bootstrap' parameter decides if bootstrap samples are used when building trees or if the whole dataset is used to build each tree while the former is the default setting. The 'criterion' parameter decides the quality of the split with an error function either 'mse', which stands for Mean Squared Error or 'mae', which stands for Mean Absolute Error, with a default value of 'mse' <sup>2</sup>.

Similar to the Random Forest, Gradient Boosting also has many parameters out of which the focus is on 'max\_features', 'learning\_rate', 'criterion', 'loss', and 'n\_estimators'. The 'max\_features', 'n\_estimators' and 'criterion' have the same function and possible values as for the random forests. The 'learning\_rate' shrinks the contribution of each constituent tree by the value held by learning\_rate whose default value is 0.1. The 'loss' parameter decides the loss function to be optimized and has the possible values 'ls' referring to least squares regression, 'lad' which stands for least absolute deviation, 'huber' which is a combination of the former two and 'quantile' which allows quantile regression. The default value for this parameter is 'ls' <sup>3</sup>.

### **Neural Network model**

---

<sup>2</sup>For more details on the parameter values, refer to: [RandomForestRegressor](#)

<sup>3</sup>[GradientBoostingRegressor](#)

The neural network model used in this study is the multi-layer perceptron (MLP). Out of the possible parameters that can be tuned, the focus of this study is on the parameters 'activation', 'solver', 'alpha', and 'learning\_rate'. The 'activation' parameter defines the activation function for the each neuron in a hidden layer and can be 'identity', 'logistic', 'tanh' or 'relu' which represent different mathematical functions with the default one being 'relu'. The 'solver' is used for weight optimization and can be among 'lbfgs', 'sgd' and 'adam' with the default value being 'adam' which refers to a stochastic gradient-based optimizer. 'alpha' is the regularization penalty and is usually 0.0001. The 'learning\_rate' is the rate at which weights are updated and can be among 'constant', 'invscaling', and 'adaptive' with the default value being 'constant' <sup>4</sup>.

### Chosen Parameters

All eight models were built from the optimal parameters chosen from the list of values with respect to minimum error, resulting from the grid search function. The best values for each parameter of each model for both the spectral and abundance frameworks are listed in table 5.1.

## 5.3 Model Validation

The models developed with the chosen parameters from section 5.2 need to be validated. Model validation is the process of verifying if a model performs as it is intended to perform. For a regression model, validation can involve analyzing the goodness of fit, checking if the regression residuals are random, performing an out-of-sample testing.

### 5.3.1 Regression Residuals

The residual of a model is the difference between the true values and the predicted values from a regression model. Generally, a regression model is said to have performed as expected if its residuals are normally distributed and centered around zero while a random distribution means the model performance is below average.

Mathematically, residuals can be calculates as

$$\text{residuals} = \text{truevalue} - \text{predictedvalue}. \quad (5.1)$$

---

<sup>4</sup>MLPRegressor

**Table 5.1:** The best parameters from the Grid Search for all the ML regression models

	Parameter	Spectral	Abundance
<b>Linear Models</b>			
Linear Regression	none	none	none
Ridge Regression	alpha	15	30
	solver	lsqr	sag
Partial Least Squares	none	none	none
<b>Non-parametric Models</b>			
K-Nearest Neighbors	weights	uniform	uniform
	algorithm	auto	auto
Support Vector Machines	kernel	rbf	rbf
	degree	0	0
	gamma	0.001	scale
	tolerance	0.001	0.0001
	C	0.001	0.1
<b>Ensemble Models</b>			
Random Forest	max_features	sqrt	log2
	n_estimators	150	50
	bootstrap	True	True
	criterion	mse	mse
Gradient Boosting	max_features	sqrt	auto
	learning_rate	0.1	0.001
	criterion	mae	mse
	loss	huber	lad
	n_estimators	10	150
<b>Neural Model</b>			
Multi-layer Perceptron	activation	identity	logistic
	solver	lbfgs	sgd
	alpha	0.1	0.0001
	learning_rate	adaptive	adaptive

When the distribution is skewed<sup>5</sup> towards the left with more density lesser than zero, the model is said to predict values higher than the true values whereas with more density greater than zero meaning, the model predicts values lesser than the true values.

### 5.3.2 Goodness of fit

The coefficient of determination, also called the R-squared score ( $R^2$ ) is a measure to explain the correlation between dependent and independent variables in a regression analysis. It can be explained as a proportion of the variance in the dependent variable that is predictable from the independent variable. It is also known as the 'goodness of fit' measure, with a maximum value of 1 which indicates a perfect fit, while a value of 0 would indicate that the calculation fails to accurately model the data as it uses the

<sup>5</sup>For more in detail, refer to: [skewness](#)

arithmetic mean as the predicted value and any value below 0 indicates an arbitrarily worse model. It should be noted that a negative value for the R-squared score indicates negative correlation of the predicted values to the true values.

Mathematically, it can be calculated as,

$$R^2 = 1 - \frac{RSS}{TSS} \quad (5.2)$$

where  $r^2$  is the coefficient of determination,  $RSS$  is the sum of squares of residuals and  $TSS$  is the total sum of squares.

### 5.3.3 Cross-Validation

The out-of-sample testing, also called as **cross-validation** is a way of assessing how the results of a regression model will generalize to an independent data set. It can be thought of as testing if a model's predictive performance deteriorates substantially when applied to data that were not used in model estimation. Learning the parameters of a prediction function and testing it on the same data is an is not recommended and can be avoided by keeping a part of the training set unknown (unseen data) to the model and training it with only the known data (learning data). The model is tested against the unseen data to assess its performance.

The goal of cross-validation is to test the model's ability to predict new data that was not used in training it, which helps avoid overfitting or underfitting. It also gives an insight on how the model will generalize to an independent dataset. Some well known validation techniques include *K-Fold cross validation*, *holdout method*, and *Leave-One-Out cross validation*.

#### **Leave One Out Cross Validation (LOO-CV)**

Leave One Out Cross Validation (LOO-CV) is a simple cross-validation technique in which each learning data set is created by taking all the training samples except one. The unknown data in this case, is the sample that has been left out. This ensures for every iteration, the learning data set and unknown data are different while simultaneously ensuring a larger set of training data. In this study, I have chosen to implement the **Leave One Out Cross Validation (LOO-CV)** technique into the grid search for the parameter tuning. Employing this validation technique for parameter tuning ensures the parameters and models thus developed do not overfit.

## 5.4 Model Evaluation

Model evaluation is the process of assessing or estimating a model's performance. It must be noted that the term accuracy is a measure for classification and not regression, and hence measures like accuracy, precision or recall cannot be used to evaluate the regression models in this study. The performance of regression models can be determined using certain types of evaluation metrics like Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE).

### 5.4.1 Evaluation metrics

The metrics used in this study are discussed in this section. The scikit-learn library provides built-in functions for these metrics which has been used in this study. MSE, RMSE and MAE are measured relatively to a specific dataset and a lower value indicates better model performance.

#### Mean Squared Error (MSE)

Mean Squared Error (MSE) is a widely used error metric and an important loss function for algorithms using the least squares for regression analysis. The linear models discussed in section 2.5 which use the 'least squares' method refers to minimizing the mean squared error between the predicted values and expected values.

Mathematically, it is the mean of the squared differences between predicted and expected values and can be represented as,

$$MSE = \frac{1}{n} \sum_{t=1}^n (y_t - \hat{y}_t)^2 \quad (5.3)$$

where  $n$  is the number of data points,  $y_t$  is the expected value and  $\hat{y}_t$  is the predicted value. The difference between these two values is squared, which has the effect of removing the sign, resulting in a positive error value and also has the effect of magnifying large errors, thus MSE does not have the same unit of the target value.

#### Root Mean Squared Error (RMSE)

Root Mean Squared Error (RMSE) is an extension of MSE. It can be calculated as the square root of the errors calculated and is represented as,

$$RMSE = \sqrt{\frac{1}{n} \sum_{t=1}^n (y_t - \hat{y}_t)^2} \quad (5.4)$$

where,  $n$  is the number of data points,  $y_t$  is the expected value and  $\hat{y}_t$  is the predicted value or as,

$$RMSE = \sqrt{(MSE)} \quad (5.5)$$

It should be noted that RMSE cannot be calculated as the average of the square root of the MSE values but as the square root of the entire mean squared error. The units of the error score match the units of the target value that is being predicted as the squaring in the MSE is reversed by applying a square root in the RMSE.

### **Mean Absolute Error (MAE)**

Mean Absolute Error (MAE) like RMSE has the units of the target value but unlike the RMSE, the changes in MAE are linear and therefore intuitive. The MAE does not give different weight to different types of errors. Instead the scores increase linearly with increases in error much unlike the MSE and RMSE which inflate the mean error score.

The MAE score is calculated as the average of the absolute error values and can be represented as,

$$MAE = \frac{1}{n} \sum_{t=1}^n |y_t - \hat{y}_t| \quad (5.6)$$

where,  $n$  is the number of data points,  $y_t$  is the expected value and  $\hat{y}_t$  is the predicted value.

Taking the MAE forces a positive error value, independent of the sign of the deviation meaning independent of whether the predicted or expected value is larger.





## Chapter 6

# Results and Discussion

This chapter discusses the results of the experiments performed for this study which includes the results from the regression analysis of the eight models for each of the spectral and abundance frameworks. The model-wise results of both the frameworks are discussed in detail in section 6.1, followed by the overall results for both frameworks discussed in section 6.2. The process of product integration is discussed in section 6.4 followed by the limitations in section 6.5.

### 6.1 Model-wise Results

The results from each model from each category are discussed in detail in this section and a comparison is drawn between their performance on the spectral data and abundance data. This gives a better understanding of how the raw spectral data which has been extracted from sampling locations and the abundance data obtained from constrained spectral unmixing contribute for estimating the concentration of blood in a salmon fillet.

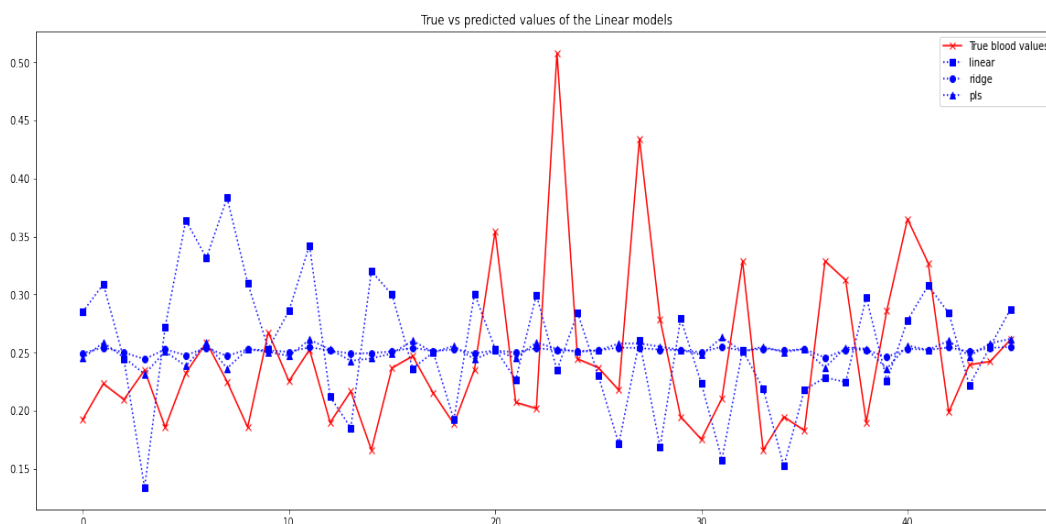
LOO-CV as presented in section 5.3 is applied on the grid search for all models. Their corresponding best score values for all models have been collected. This score is the mean cross-validated score (CV score) of the best estimator and gives a relative idea about the performance of a model on unseen data.

The residuals on the test set as presented in section 5.3, have been calculated for every model and their distribution is plotted as a histogram. Along with this, the goodness of fit measure, also called the R-squared score as presented in section 5.3 is calculated for every model. The evaluation metrics discussed in section 5.4 like the MSE, RMSE and MAE are calculated for every model and are discussed in detail in this section.

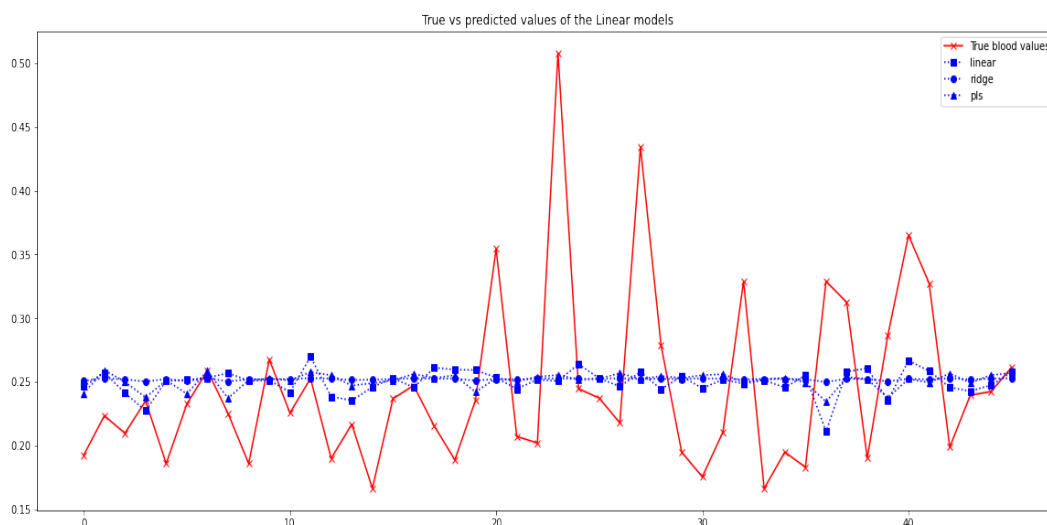
### 6.1.1 Linear model

The linear models used in this study are linear regression, ridge regression and partial least squares regression. The plot of the true values vs predicted values from the three linear models on the spectral framework can be seen in figure 6.1 and on the abundance framework can be seen in figure 6.2.

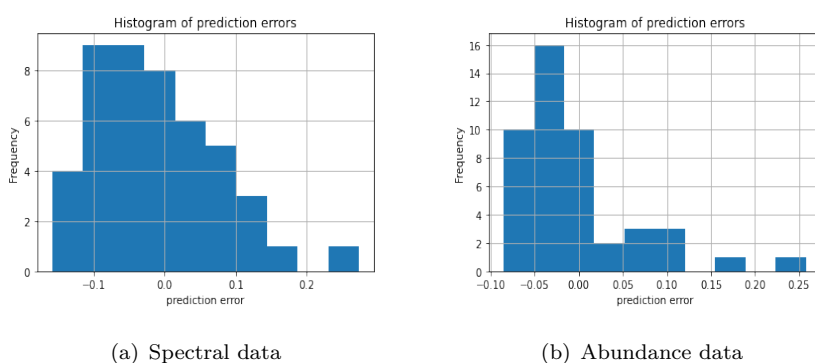
**Figure 6.1:** Plot of the linear regression models on the spectral framework



The solid red line in both the plots represents the true blood values and the other lines represent each of the models and the dashed blue lines represent the models. Each model is depicted by a different symbol depicting the data point, as listed in the plot legend. Observing the plot in figure 6.1, it can be seen that the predictions from the linear regression are erratic and performs worse than the ridge and PLS regression models. This behaviour can be attributed to the high number of features in the spectral framework, i.e., 216 features. While observing figure 6.2, it can be seen that this is not the case with linear regression on abundance data. The reason for this could most likely be that the abundance data only has 9 features and hence the performance of linear regression is similar to that of ridge and PLS regression. The performance of ridge and PLS regression are similar on both the spectral and abundance data.

**Figure 6.2:** Plot of the linear regression models on the abundance framework

The histogram distribution of the residuals from each model for both the frameworks are presented in figures 6.3, 6.4, and 6.5.

**Figure 6.3:** Histogram of residual error distributions from Linear Regression.

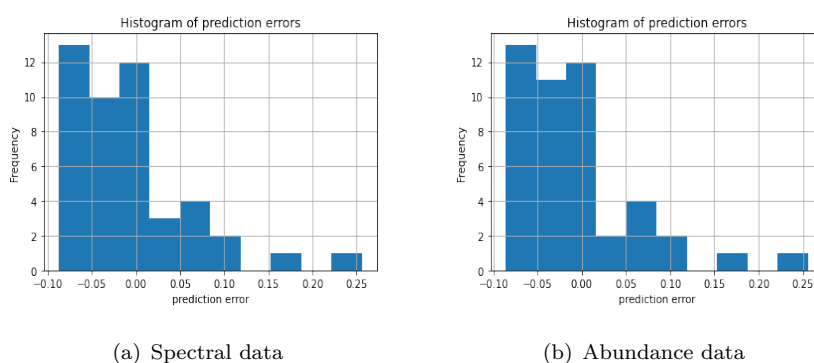
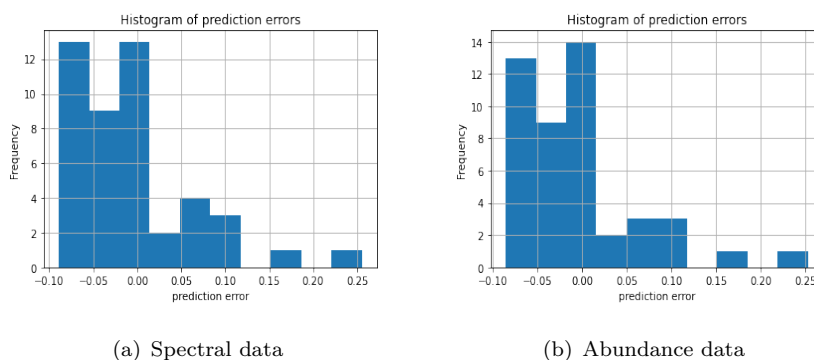
Observing figure 6.3, the linear model predicts values higher and lower than the true values for both the spectral and the abundance data, while the same is true for ridge regression and PLS regression, whose residual error distributions can be seen in figure 6.4 and figure 6.5.

**Table 6.1:** Scores and Errors from each Linear model for spectral data

	MSE	RMSE	MAE	$R^2\_score$	CV_score
Linear Regression	0.007806	0.088354	0.071814	-0.6469	-0.0047
Ridge Regression	0.004781	0.069147	0.051648	-0.0087	-0.0026
Partial Least Squares	0.004766	0.069035	0.051134	-0.0054	-0.0026

**Table 6.2:** Scores and Errors from each Linear model for abundance data

	MSE	RMSE	MAE	$R^2\_score$	CV_score
Linear Regression	0.0048	0.0691	0.0515	-0.0080	-0.0027
Ridge Regression	0.004786	0.06918	0.051859	-0.0097	-0.00262
Partial Least Squares	0.004823	0.069449	0.051455	-0.0175	-0.00278

**Figure 6.4:** Histogram of residual error distribution from Ridge Regression.**Figure 6.5:** Histogram of residual error distribution from PLS Regression.

The R-squared score and CV score along with the evaluation metrics from the three models for the spectral data are listed in table 6.1 and for the abundance data are listed in table 6.2

Based on the values from tables 6.1 and 6.1, the error metrics MSE, RMSE and MAE are minimum for PLS regression on spectral data whereas both linear regression and PLS regression have slightly better error values on the abundance data. The  $R^2$  score

and CV score for PLS regression on spectral data shows it as the relatively better linear model whereas on abundance data, both linear regression and PLS regression appear to be performing slightly better. All three models are pretty similar, except the R2 is a bit worse on the PLS.

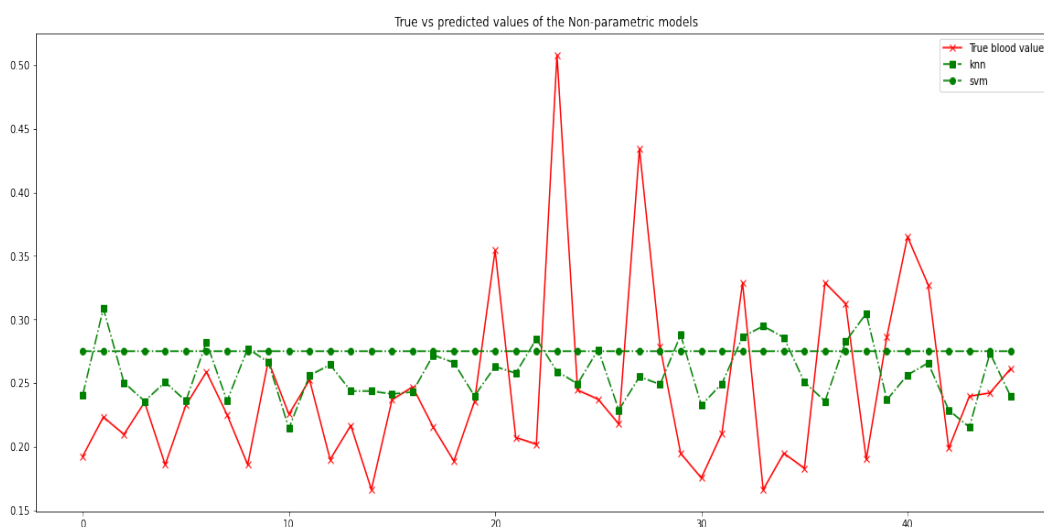
But the plots in figures 6.1 and 6.2 show that the PLS model on spectral data and the linear and PLS models on abundance data seem to be mostly straight lines and do not capture the nuances of either data as they should. This can be logically understood as the linear models work on minimising the error between the predicted and true value for which, when the true values represented by the continuous red line in the plots, a nearly straight line can be equidistant from the true values, thus having straight line.

The MAE and RMSE have the same units as the data. They are around 0.7 mg HB / g muscle, which is roughly what one would get by using the mean of the training data as an estimator for all the points. This analysis thus shows that the values of the metrics can be used to quantify a model relative to another but does not give more information on a standalone model's performance.

### 6.1.2 Non-parametric model

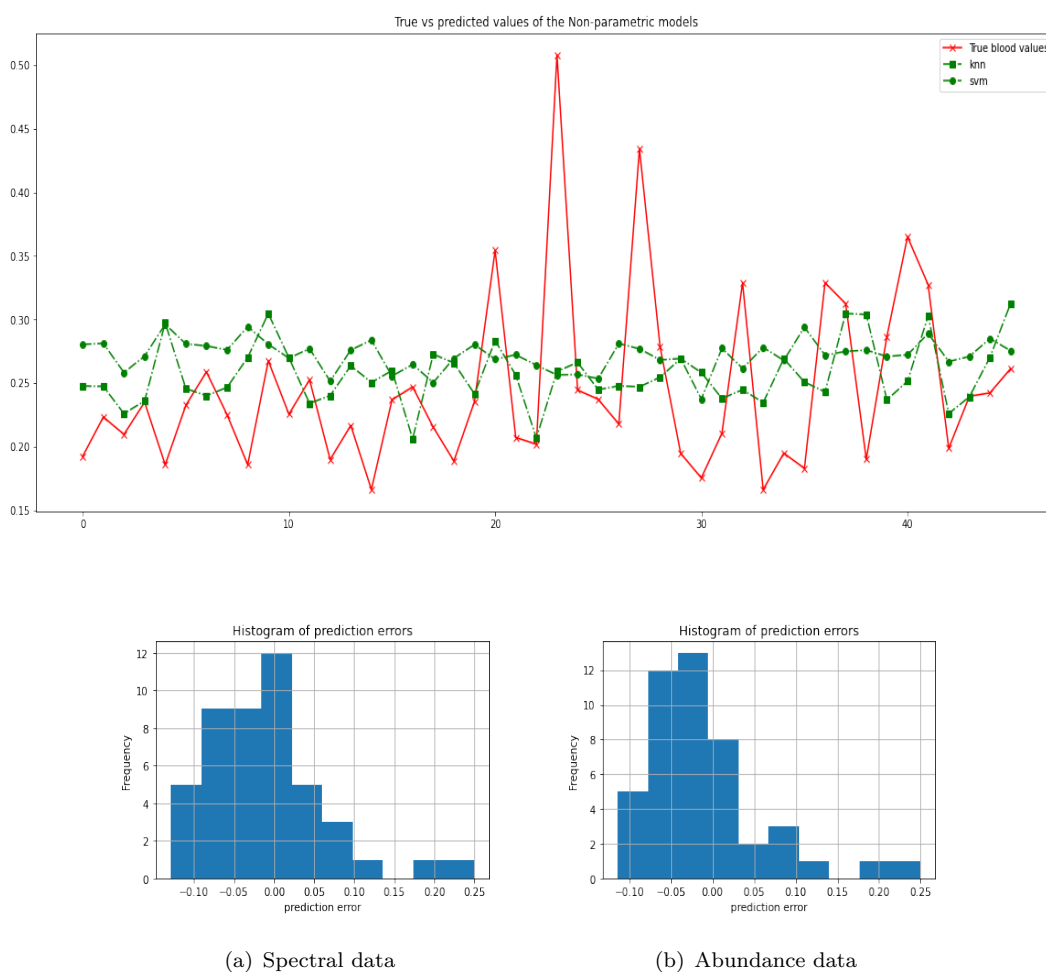
The non-parametric models used in this study are K Nearest neighbors (KNN) and Support Vector Machines (SVM) and the plot of true values vs predicted values from the two non-parametric models on the spectral framework can be seen in figure 6.6 and on abundance framework can be seen in figure 6.7.

**Figure 6.6:** Plot of the non-parametric regression models on the spectral framework



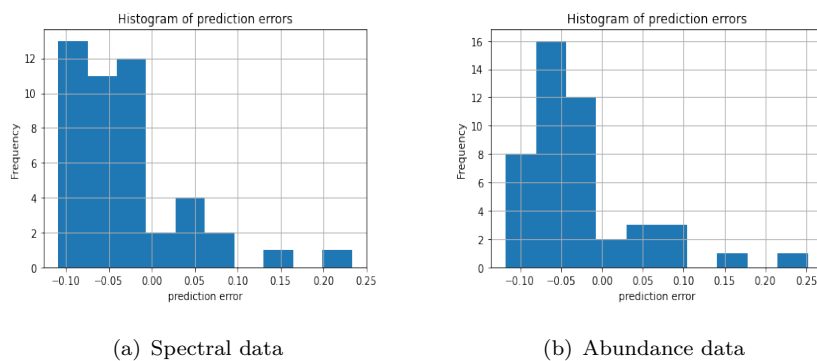
The solid red line in both the plots represents the true blood values and the other lines represent each of the models and the dashed green lines represent the models. Each model is depicted by a different symbol depicting the data point, as listed in the plot legend. Observing the plot in figure 6.6, the predicted values from SVM is a straight line which is unusual considering the kernel chosen from grid search discussed in section 5.2.1 for the spectral data is Radial Basis Function or 'rbf'. It is a kernel which computes the similarity of data points to each other. This is not the case on abundance data, as can be seen in figure 6.7. This can again be explained by the number of features in each model and since in both cases, the model chooses the parameter that gives out minimal error, the number of features in spectral data may be the cause of this issue. KNN on the other hand performs similar on both the spectral and abundance data.

**Figure 6.7:** Plot of the non-parametric regression models on the abundance framework



**Figure 6.8:** Histogram of residual error distribution from K-Nearest neighbors.

The histogram distribution of the residuals from each model for both the frameworks are presented in figures 6.8 and 6.9.



**Figure 6.9:** Histogram of residual error distribution from Support Vector Machines.

The error distribution of SVM from figure 6.9 shows the negative skewness for both the frameworks. This could be because the data has some very high values that increase the mean, while most points are under the mean, which applies to KNN from figure 6.8. While this gives an idea about both the models and how their predicted values are more often higher than the true values, it fails to explain more about the characteristic attributes of the models.

**Table 6.3:** Scores and Errors from each Non-parametric model for spectral data

	<b>MSE</b>	<b>RMSE</b>	<b>MAE</b>	$R^2$ <b>_score</b>	<b>CV_score</b>
K-Nearest Neighbors	0.005418	0.073607	0.055077	-0.143	-0.0035
Support Vector Machines	0.005631	0.075043	0.063517	-0.1881	-0.0033

The R-squared score and CV score along with the evaluation metrics from the two models for the spectral data are listed in table 6.3 and for the abundance data are listed in table 6.4.

**Table 6.4:** Scores and Errors from each Non-parametric model for abundance data

	<b>MSE</b>	<b>RMSE</b>	<b>MAE</b>	$R^2$ <b>_score</b>	<b>CV_score</b>
K-Nearest Neighbors	0.00508	0.071277	0.053453	-0.0718	-0.00317
Support Vector Machines	0.005765	0.075931	0.06222	-0.2163	-0.00318

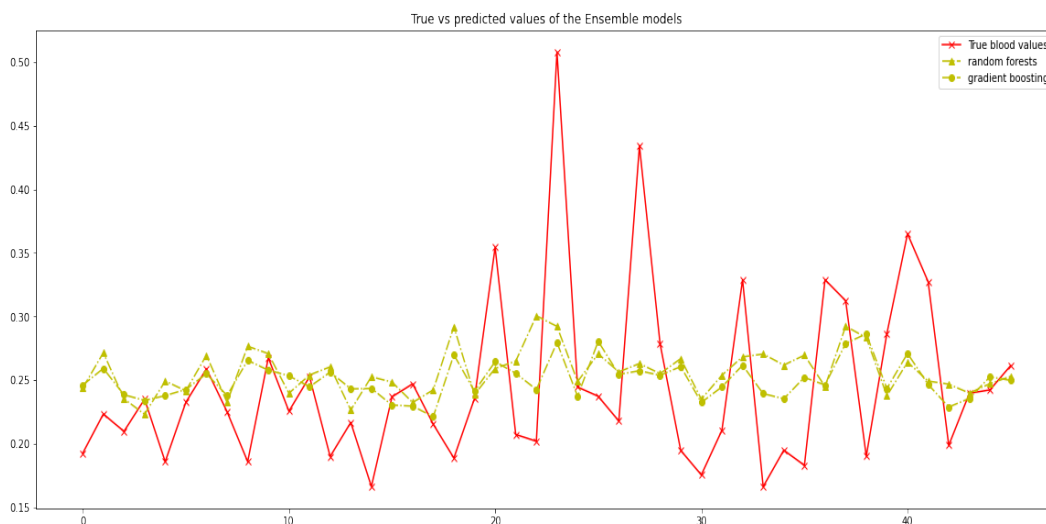
Based on the values from tables 6.3 and 6.4, the error metrics MSE, RMSE and MAE among the models are minimal for KNN regression on both the spectral data and abundance data. The  $R^2$  score and CV score for both KNN and SVM regression on spectral data are close whereas on the abundance data, KNN is significantly better performing. This could be due to KNN being simpler and thus more robust.

The plots in figures 6.6 and 6.7 show that the SVM on spectral data would not be a good fit since the flatter the line, the better the model if it has decided to use the mean as an estimator. Whereas on abundance data, it is all over the place and looks like it could be better but the metrics conflict with this, while the error distribution of SVM looks skewed for both the spectral and abundance data. All this clearly shows that these metrics might not be enough to evaluate the performance of a standalone SVM model. While in comparison with KNN which is clearly better in both the metrics and error distribution, an understanding can be made as to which of the two chosen non-parametric models perform better on the data at hand. This analysis thus shows again that the values of the error metrics does not give more information on the model's performance and thus must rely on  $R^2$  and CV scores as well.

### 6.1.3 Ensemble model

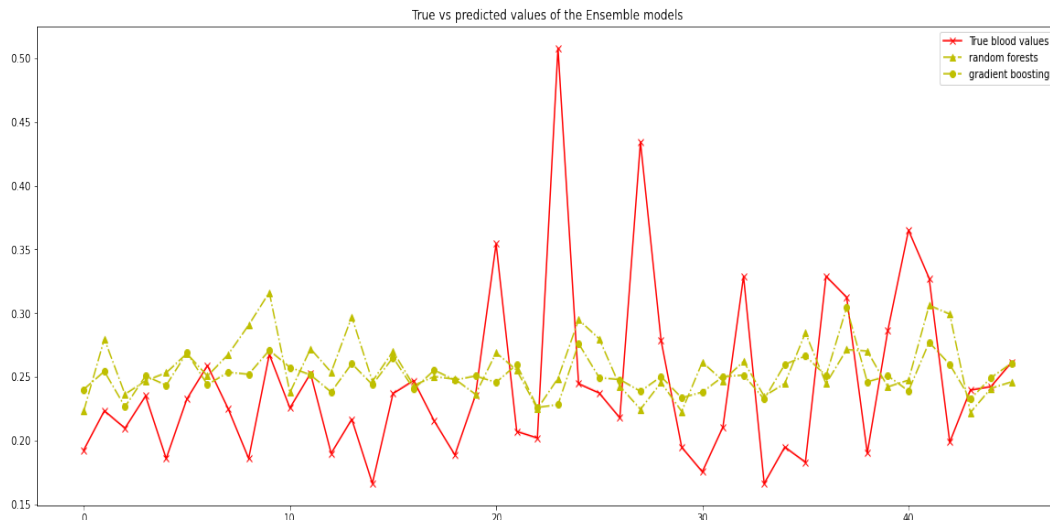
The ensemble models used in this study are Random Forests (RF) and Gradient Boosting (GB). The plot of true values vs predicted values from the two ensemble models on the spectral framework can be seen in figure 6.10 and on abundance framework can be seen in figure 6.11.

**Figure 6.10:** Plot of the ensemble regression models on the spectral framework

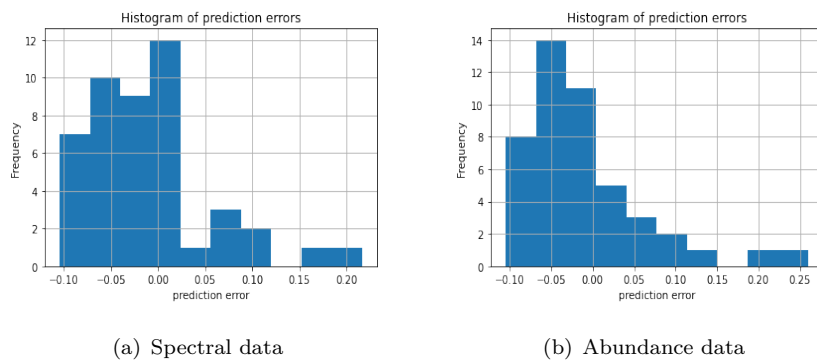


Observing the plots in figures 6.10 and 6.11, both the models predict values which in some sense follow the trend of the true values while not giving a clearer picture about their relative performance as seen in the case of linear and non-parametric models above.

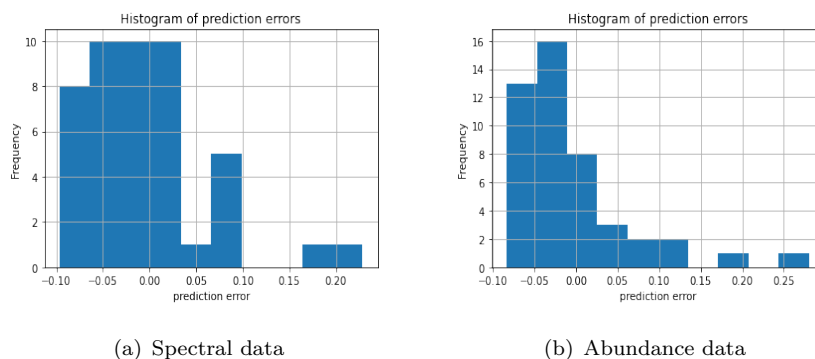


**Figure 6.11:** Plot of the ensemble regression models on the abundance framework

The histogram distribution of the residuals from each model for both the frameworks are presented in figures 6.12, and 6.13.

**Figure 6.12:** Histogram of residual error distribution from Random Forest.

The error distribution of Random Forests from figure 6.12 shows negative skewness for both spectral and abundance data which is similar to the Gradient Boosting from figure 6.13. Both the error distribution of both models on the abundance data seems more normal as compared to the spectral data. Nevertheless, these plots show that both models predict values higher than the true values.



**Figure 6.13:** Histogram of residual error distribution from Gradient Boosting.

The  $R^2$  score, and CV score along with the evaluation metrics from the two models for the spectral data are listed in table 6.5 and for the abundance data are listed in table 6.6.

**Table 6.5:** Scores and Errors from each Ensemble model for spectral data

	MSE	RMSE	MAE	$R^2\_score$	CV_score
Random Forest	0.004752	0.068935	0.051812	-0.0025	-0.0027
Gradient Boosting	0.00421	0.064885	0.047843	0.1118	-0.0025

**Table 6.6:** Scores and Errors from each Non-parametric model for abundance data

	MSE	RMSE	MAE	$R^2\_score$	CV_score
Random Forest	0.005449	0.073819	0.055642	-0.1496	-0.00275
Gradient Boosting	0.004944	0.070312	0.049909	-0.43	-0.002526

Based on the values from tables 6.5 and 6.6, the error metrics MSE, RMSE and MAE are minimum for Gradient Boosting regression on both the spectral data and abundance data. The  $R^2$  score of Gradient Boosting is significantly higher than random forests on the spectral data while it is the other way round on the abundance data. This uncharacteristic behaviour can be attributed to the chosen parameters in table 5.1 in section 5.2.1. Specifically the number of estimators of gradient boosting on the abundance data is much higher than that for spectral data. The individual parameter estimation cannot be explained when using grid search as these best parameters are chosen as the combination of values which result in least error.

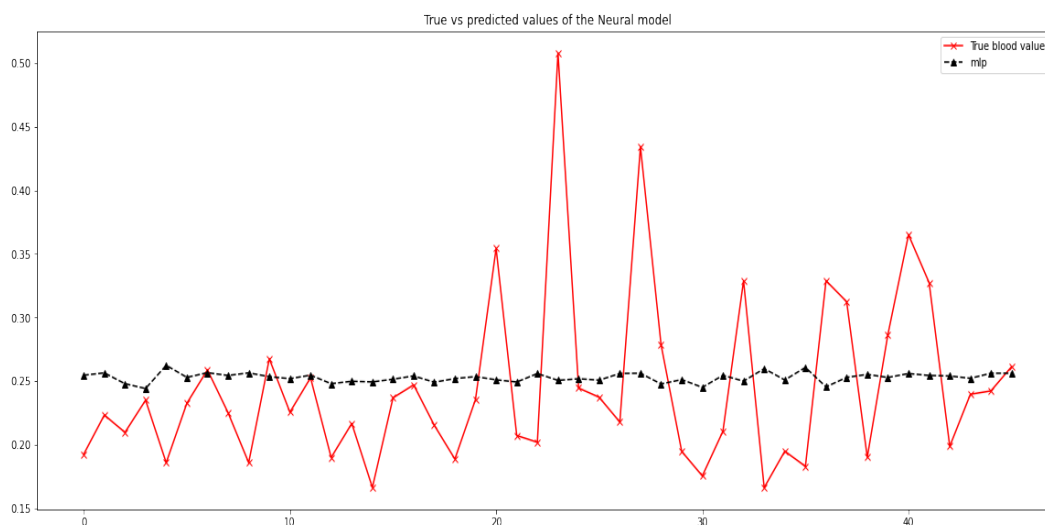
The CV scores of gradient boosting seem similar in both frameworks which shows the chosen parameters on cross validation have performed as expected. The random forests model for both frameworks perform similarly and relatively not better than the gradient boosting model.

While the plots in figures 6.10 and 6.11 give a bit of an idea about their performance, the evaluation metrics explain better. Although gradient boosting model is clearly better than random forests on the spectral data, it is not the case for the abundance data. This can be explained as the lack of enough features given the number of estimators. The issue with the performance of other linear and non-parametric models on spectral data was the high number of features while it is a plus while using gradient boosting. Simpler models perform better on abundance data with less number of features as expected. This cannot be generalised as the parameters were tuned to this specific dataset and may not apply to a larger or different data.

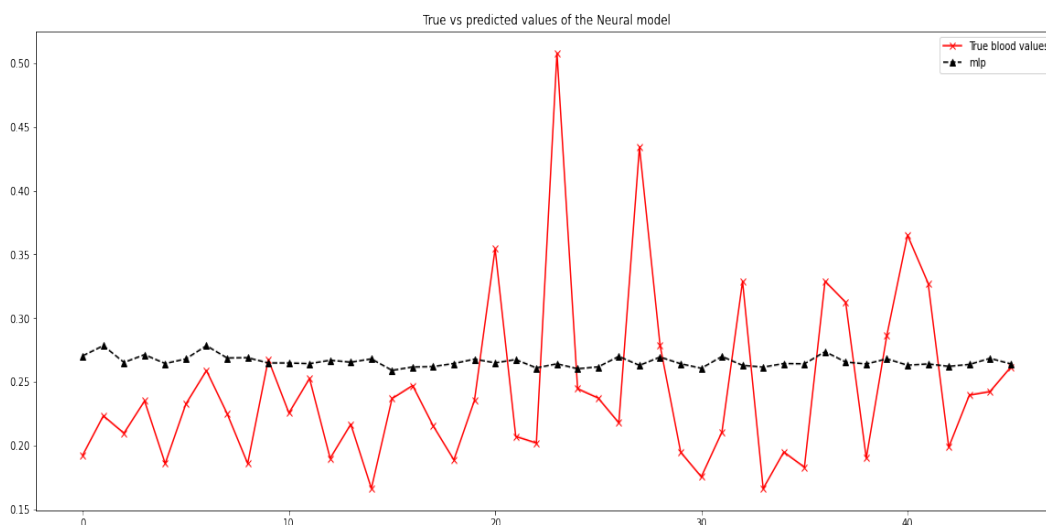
### 6.1.4 Neural model

The neural model used in this study is the multi-layer perceptron (MLP) and the plot of true values vs predicted values from the neural model on the spectral framework can be seen in figure 6.14 and on the abundance framework can be seen in figure 6.15

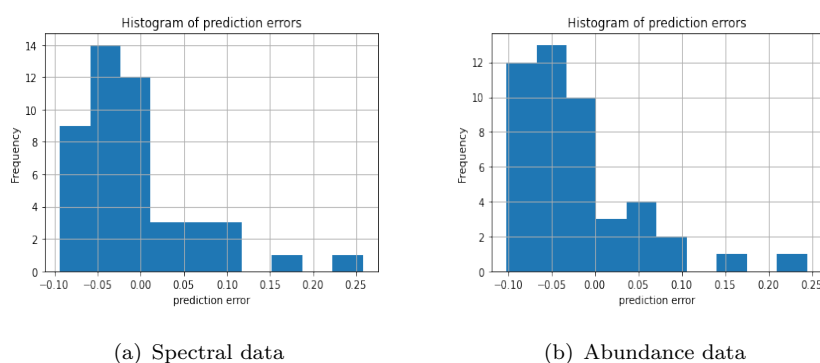
**Figure 6.14:** Plot of the neural regression model on the spectral framework



Observing both plots in figures 6.14 and 6.15, the MLP finds an approximate fit on both the data that is not necessarily a straight line nor a complex function, where the solid red line in both the plots represent the true blood values and the other line represents the predicted values from the MLP model. This can again be attributed to the process of parameter tuning by minimising the error. A non-linear function which can approximate the best distance from the true value points based on minimal error would look like the lines in these plots. The model seems to think that the best estimator is a straight line through the mean, so although it is a non-linear model, it yields a simple estimator, probably the due to the size of the training data.

**Figure 6.15:** Plot of the neural regression model on the abundance framework

The histogram distribution of the residuals from the MLP model for both the frameworks are presented in figure 6.16.

**Figure 6.16:** Histogram of residual error distribution from Multi-Layer Perceptron.

The error distribution of MLP from figure 6.16 shows negative skewness for both the spectral and abundance data and the errors are not normally distributed. MLP also predicts values higher than the true values and does not capture the nuances of either data. It can seem surprising for a neural model to fail in capturing data nuances but in this case, only a very basic neural model was employed and the parameters chosen were to get minimal error, given the limited number of data points. This can be thought of as underfitting but one cannot know more without looking at the evaluation metrics.

**Table 6.7:** Scores and Errors from each Neural model for spectral data

	MSE	RMSE	MAE	$R^2\_score$	CV_score
Multi-layer Perceptron	0.00486	0.069717	0.052362	-0.0254	-0.00254

**Table 6.8:** Scores and Errors from each Non-parametric model for abundance data

	MSE	RMSE	MAE	$R^2\_score$	CV_score
Multi-layer Perceptron	0.0052	0.0721	0.0584	-0.0967	-0.0025

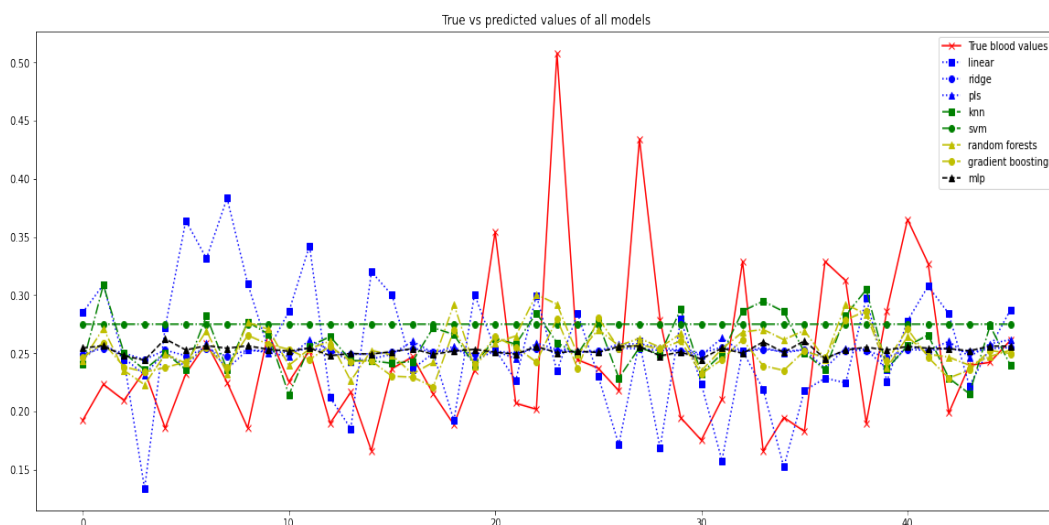
The  $R^2$  score, and CV score along with the evaluation metrics from the MLP model for the spectral data are listed in table 6.7 and for the abundance data in table 6.8.

The MLP model has slightly lower error values on the spectral data than on the abundance data which also applies to the  $R^2$  score and CV scores, which shows that MLP performs slightly better with high number of features much like the gradient boosting model. It has been established in the above subsections that the evaluation metrics can help understand the relative performance of a model. So, the metrics of MLP can be considered when being compared to either the non-parametric models or the ensemble models. Keeping that in mind and observing the tables 6.7 and 6.8, the errors of MLP are lower than that of the non-parametric models but not the ensemble models while the  $R^2$  score and CV score are almost similar to that of the ensemble and non-parametric models on both spectral and abundance data.

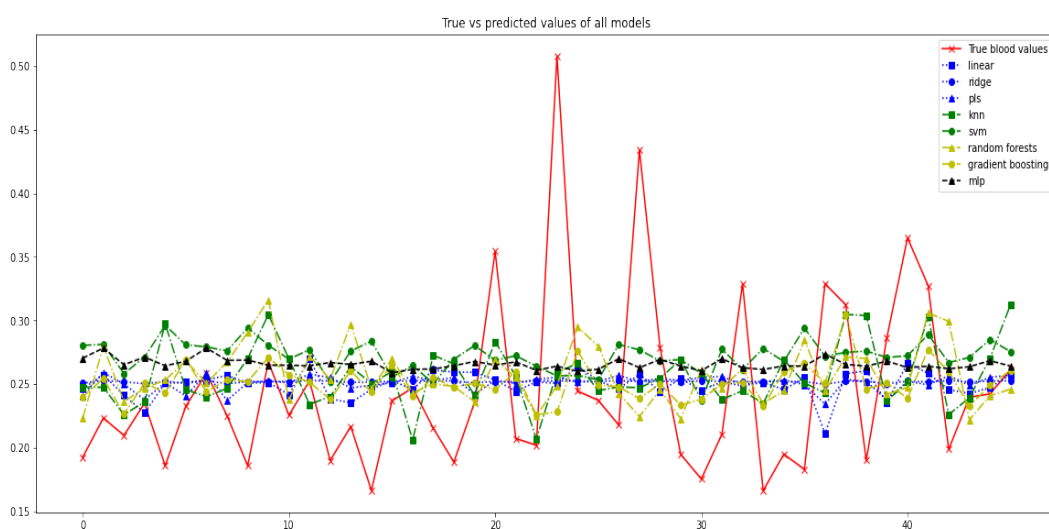
## 6.2 Overall Results

The results from the regression analysis of the eight machine learning models developed, validated and evaluated in chapter 5 and individually discussed in the subsections of section 6.1 are summarised in this section.

The plot of the predictions vs true values from all eight models for the spectral framework is presented in figure 6.17 and for the abundance framework is presented in figure 6.18.

**Figure 6.17:** Plot of the regression models on the spectral framework

The summary plot in figure 6.17 for the spectral data does not give much of a new insight as to the performance of the models. But it can be observed that except the linear estimation models, all other models have chosen a functions which can be thought of as a nearly linear estimation of the true values, approximately a straight line. While in the plot in 6.18 for the abundance data, all models follow a nearly linear estimation including the linear regression and as stated in the section 6.1, this can be attributed to the number of features.

**Figure 6.18:** Plot of the regression models on the abundance framework

The overall  $R^2$  scores and CV scores along with the evaluation metrics like MSE, RMSE and MAE for each individual model have been summarised in table 6.9 for the spectral data and in table 6.10 for the abundance data.

**Table 6.9:** Scores and Errors from each models for spectral data

	<b>MSE</b>	<b>RMSE</b>	<b>MAE</b>	<b><math>R^2</math>_score</b>	<b>CV_score</b>
Linear Regression	0.007806	0.088354	0.071814	-0.6469	-0.0047
Ridge Regression	0.004781	0.069147	0.051648	-0.0087	-0.0026
Partial Least Squares	0.004766	0.069035	0.051134	-0.0054	-0.0026
K-Nearest Neighbors	0.005418	0.073607	0.055077	-0.143	-0.0035
Support Vector Machines	0.005631	0.075043	0.063517	-0.1881	-0.0033
Random Forest	0.004752	0.068935	0.051812	-0.0025	-0.0027
Gradient Boosting	0.00421	0.064885	0.047843	0.1118	-0.0025
Multi-layer Perceptron	0.00486	0.069717	0.052362	-0.0254	-0.00254

The evaluation metrics as presented in section 6.1, can only give an idea about the performance of the models relative to one another. All the models struggle, but some manage better than others. With more data or data augmentation they would work better. The error scores of the models on spectral data from table 6.9 are all nearly close to each other but **gradient boosting** has the minimum error in all three, MSE, RMSE and MAE. While comparing the  $R^2$  and CV scores, gradient boosting performs significantly better than the other models on the spectral data. This is followed by random forests which is also an ensemble model. Thus, it suffices to say that high number of features in the spectral model are relatively better captured by the ensemble models.

**Table 6.10:** Scores and Errors from each models for abundance data

	<b>MSE</b>	<b>RMSE</b>	<b>MAE</b>	<b><math>R^2</math>_score</b>	<b>CV_score</b>
Linear Regression	0.0048	0.0691	0.0515	-0.0080	-0.0027
Ridge Regression	0.004786	0.06918	0.051859	-0.0097	-0.00262
Partial Least Squares	0.004823	0.069449	0.051455	-0.0175	-0.00278
K-Nearest Neighbors	0.00508	0.071277	0.053453	-0.0718	-0.00317
Support Vector Machines	0.005765	0.075931	0.06222	-0.2163	-0.00318
Random Forest	0.005449	0.073819	0.055642	-0.1496	-0.00275
Gradient Boosting	0.004944	0.070312	0.049909	-0.43	-0.002526
Multi-layer Perceptron	0.0052	0.0721	0.0584	-0.0967	-0.0025

A similar analysis of the model metrics for the abundance data in table 6.10 shows that error-wise, the linear regression has least mean squared errors (MSE and RMSE) while gradient boosting has the least mean absolute error (MAE). By definition, MAE is known to be less biased for higher values and may not adequately reflect the performance of a model when dealing with large error values. While MSE is highly biased for higher values and RMSE is better in terms of reflecting performance when dealing with large error

values. Based on this definition of the errors, and the fact that the difference between the values of linear regression and gradient boosting are significantly very close, both models seem to be capturing the abundance data better than the spectral data. It can be said that both models fail to capture the data perfectly well, since the mean errors are close to the average distance between the data points and the mean, and the  $R^2$  and CV scores are close to zero, so just using the mean as an estimator would work roughly as well.

The  $R^2$  score for the abundance data of the different models show that linear model is much better than the rest while gradient boosting has the worst  $R^2$  score. The  $R^2$  scores presented in section 5.3 depict the goodness of fit and the case of gradient boosting can be explained due to its best parameter for the number of estimators as shown in table 5.1. The number of estimators for just 9 features of gradient boosting may have led to this issue with the  $R^2$  score. Despite this issue, this was the best chosen parameter as grid search only chooses the best combination of parameters from the given values. The CV score for the abundance data look as expected and the gradient boosting model has the best score followed by the multi-layer perceptron.

On calculating the mean, MAE and RMSE values of the true blood values, they were found to be 0.2496, 0.0419 and 0.0605 respectively. Observing the MAE and RMSE values from the different models on both data types in the tables 6.9 and 6.10, some them are much closer to the MAE and RMSE values of the true blood values mentioned above. This can explain that the models which could not perform well due to lack of data simply predicted the mean value of the true values. That is why their plots look like a straight line which is almost cutting through the mean of the true data.

### 6.3 Interpretations

The results from the regression analysis on spectral and abundance data are presented in detail in the subsections of section 6.1 and summarised in section 6.2. The explanations and logical reasoning for the corresponding results have also been presented there. This section will continue to explore the other factors affecting the performance of the models and an intuitive idea behind the setup of this regression analysis.

The main objective of this thesis is to act as a foundation upon which more complex ways of detecting blood in salmon fillets can be developed. The two types of data prepared for this analysis are the raw spectral data from the pixels and the abundance data by applying CSUM on the pixels. There could be other types of data yet to be explored which can be used for this purpose.



Detecting the presence of blood in salmon has been an ongoing subject and this novel approach is an experimental study to check if the raw spectral data or the abundance data is well suited for the regression analysis. Regression analysis was chosen here with the hope to be able to predict the approximate haemoglobin concentration in a pixel from its inherent spectral or abundance information.

This thesis could have also focussed on a binary classification problem which detects if blood is present or not in a pixel. Since any salmon fillet is bound to contain some or very little strains of blood, a binary classification problem would have been cutting it too close. Whereas building a regression analysis helps in getting the blood concentration level from the pixels which in turn can be used to set a threshold value for the blood concentration and a binary classification analysis can be built on top of this to reject a fillet with concentration higher than the threshold and accept a fillet with concentration lower than the threshold.

The data pre-processing presented in detail in section 4 has multiple stages which in themselves are a research problem that needed solving. Especially the systematic approaches to automatically classifying the fillets into left and right fillet of a salmon and centreline detection in salmon fillets. The methods used in this thesis to solve these problems are now being investigated at Maritech to be integrated into the Maritech Eye product as standalone features which will help reduce the manual labor required in the salmon fillet evaluation. Apart from the exceptional industrial applications of the many steps involved in this thesis, one of the most significant contribution is that of the ridge detection in salmon fillets.

Given the many different models to choose from, the 4 categories of models chosen for this study was my choice and the results can vary on using any other different model. Also, the metrics chosen for this thesis are specific to the hyperspectral data used here and cannot be generalised for other types of hyperspectral data. Even though deep learning models were considered, they were not explored further, due to the lack of number of data instances.

Of the many ways to interpret the results from a regression analysis from the kind of comparative study performed in this thesis sheds light on how different models behave on data with different set of features and parameters. Based on an overall performance, it can be said that using **spectral data**, though computationally more expensive than abundance data, is a preferred way to go. This is because the many nuances from the 216 wavelength bands of the raw spectral data were well modelled and captured by the gradient boosting model as compared to the abundance framework for which linear regression performed best. Again, these models are far from perfect and this interpretation is *relative* and in comparison with the other models used in this study.

The `gridsearchCV` function from `scikit-learn` with LOO-CV technique was applied to ensure the selection of the best combination of parameters and thus cannot account for the individual choice of each parameter. Considering the performance of gradient boosting model on abundance data, the number of estimator parameters was too high in comparison to the number of features and might have caused the model to not perform as well as it did on the spectral data, but this could not have been foreseen as the `gridsearchCV` function chose the best performing combination and not individual parameters.

Despite the normalization and other processing of both the spectral and abundance data, none of the models seem to perform exceptionally well and this can be explained due to the very low number of data instances. This study has 113 data instances on which these machine learning models were trained. Increased number of data instances with data augmentation, could lead to better modelling in the future. This along with other ideas for future work are explored in section 7.2.

## 6.4 Product Integration

The process of product integration involves implementing the chosen model into the Maritech Eye product, which is a hyperspectral imaging setup supported by the software Breeze, developed by Prediktera, as presented in section 3.2.5 in chapter 3. The Maritech Eye product can be seen in figure 6.19.

**Figure 6.19:** The Maritech Eye product from the event launch



The model(s) to be integrated is/are chosen and converted to the ONNX format, which stands for Open Neural Network Exchange and is an open format built to represent machine learning models in various softwares. This is done using the library skl2onnx library which converts the scikit-learn models into the ONNX format which can be imported into the Breeze framework.

The workflow starts with scanning a salmon fillet using the Maritech Eye and the hyperspectral data is captured by Breeze. This includes spectral information for every pixel of the image on which a segmentation is done to separate out the fillet pixels for analysis. The CSUM is already a functionality of the product Breeze and has been integrated by Nofima and Prediktera to be applied on cod fish for blood detection. This was then optimized to be able to handle salmon as well,

The appropriate model is found from the overall results section 6.2 to be gradient boosting and can be applied on each pixel to predict the haemoglobin content in every pixel and a corresponding blood response image is built.

#### 6.4.1 Blood Response Image

The blood response image is a visual representation of the blood concentrations on different parts of a fish fillet. The blood response image from applying only the CSUM on a salmon fillet in Breeze is shown in figure 6.20.

**Figure 6.20:** The blood response image from CSUM

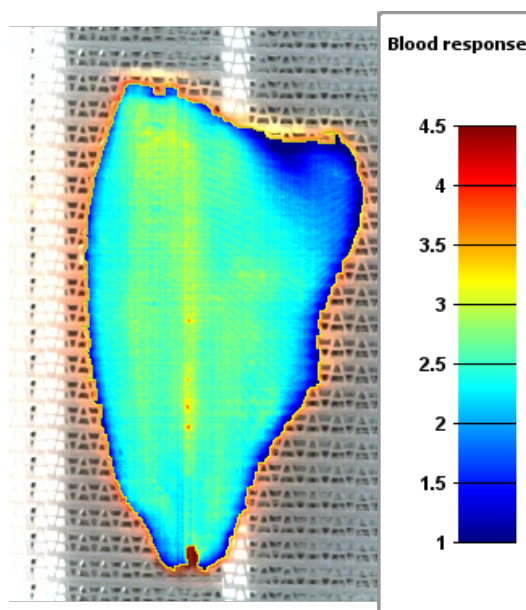


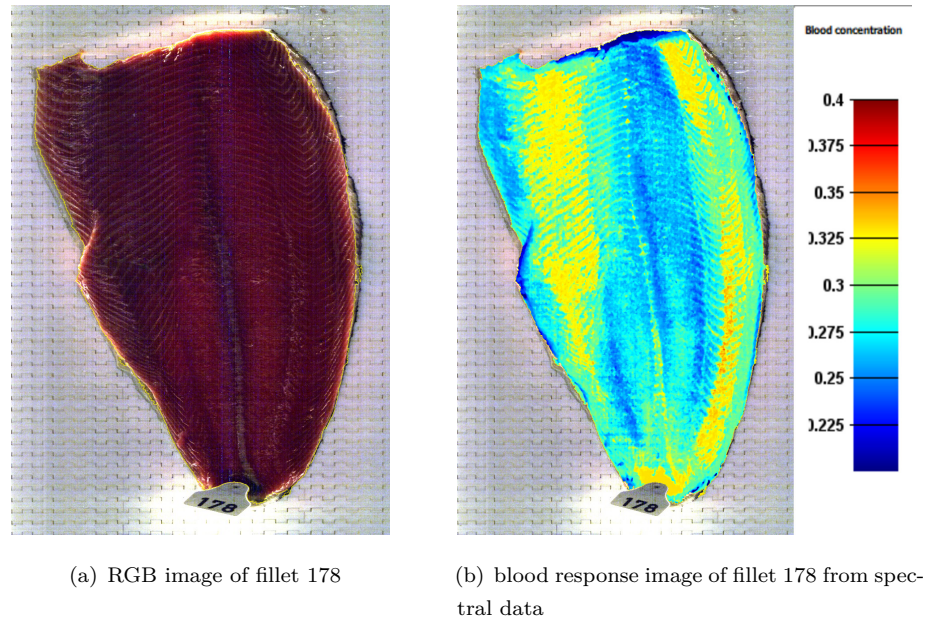
Figure 6.20 shows an example of a blood response image of fillet 177 by applying directly the CSUM, discussed in section 2.4.2. This was the method used successfully, to detect

blood in cod fish. However, the blood response image of applying this on salmon shows interference with pigments, which limits the usability for further application. The blood concentration scale on the right of the figure depicts the different levels of blood in terms of endmember abundances and thus is not very clear either.

Here, the blood spots detected using CSUM are very tiny and can be seen on the lower half of the fillet along the ridge marked in red while the rest of the fillet is relatively of the same colour. In figure 6.20, the CSUM unmixes the pixel components based on the optical properties of the endmembers present and salmon has pigments which are red in colour and have similar optical properties to that of blood, making it hard to detect the presence of blood. This makes it tricky to set proper thresholds for detecting blood spots.

Proper integration of the chosen models from section 6.2 on the spectral data with Breeze was challenging at the time of writing this thesis as Breeze is still in development stage and is a little unstable when new features are added.

The models are not usable directly on Breeze and produce an error message unless there is a certain number of training samples. This is a bug, and the developers have been informed about this and are working on this issue. Breeze expects more number of instances to train the model and lack of number of data instances was a hindrance. This issue was overcome by **data augmentation** by further splitting up the pixels from the existing sampling locations into 5x5 grids and thereby increasing the number of data instances for Breeze to train the model. Also, using only the data from the wavelengths from 555 nm up to 990 nm improved the results significantly.



**Figure 6.21:** The RGB image (a) and corresponding blood response image (b) from spectral data

Figure 6.21 shows the original vs blood response image of fillet number 178 by applying gradient boosting model on the spectral data. The blood concentration scale on the right of the figure depicts the different levels of blood concentration in terms of mg per g of muscle and is more intuitive to understand.

This is a right fillet and the presence of blood can be seen on the muscles on the right and on the left which is the belly region and the tail region along the bottom end of the fillet. These regions are expected to contain trails of blood after filleting and the blood response image depicts the same.

The improved blood response image in figure 6.21 from spectral data of salmon is more comprehensible than from figure 6.20 which was obtained by using the CSUM model. This research thus helps improve the process of filleting and evaluating the quality of a salmon fillet.

## 6.5 Limitations

The results presented in sections 6.2 are obtained using the data collected from the hyperspectral imaging setup presented in section 4.1.2 which uses halogen lamps for illumination. It was found that data in the wavelength range up to 550 nm was noisy as there was not much light from the halogen bulbs in this range and the fish muscles

scatter more light in this region. Hence considering from the wavelengths above 555 nm provided better results.

**Figure 6.22:** The noise in absorbance spectra

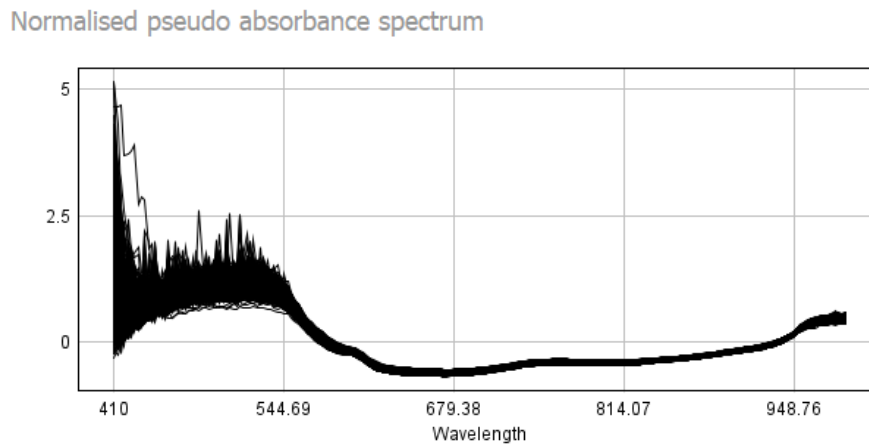


Figure 6.22 shows the absorbance spectra of the image used in Breeze during training. It can be seen that the spectra is noisy approximately between 410 nm and 555 nm. The noise is probably a combination of higher scattering of the blue light in the muscle and less blue light from the halogen bulbs. This shows it is a good idea to consider the wavelengths above 555 nm and it has proven to give improved results.

The camera in the setup, presented in section 4.1.2, used for recording these images had a higher spectral resolution and a lower signal to noise ratio (SNR) than the industrial version of the camera that is in the Maritech Eye product. The data suggests that the SNR is more important than the spectral resolution for this purpose, so the Maritech Eye is expected to perform better. It also has better light lines than the current setup meaning that the light is better focused and will therefore penetrate deeper into the sample.

## Chapter 7

# Conclusions and Future Directions

This chapter summarises the work done during the duration of this thesis and all the results obtained from the regression analysis for the detection of blood in salmon fillets. The summary of this thesis is described in section 7.1 and the scopes for future work with this thesis as a foundation is discussed in section 7.2.

### 7.1 Summary

Detecting blood in salmon fillets using hyperspectral image analysis has been achieved in this thesis using a novel approach which has not been found in use so far to the best of my knowledge. This thesis is a comparative study of different models on different types of data to find an optimal solution to the research problem. The two types of data used in this study are the raw spectral data from the pixels and the abundance data from the pixels by applying constrained spectral unmixing (CSUM). The results of the regression analysis using eight different machine learning models on both types of data as presented in chapter 6 shows that using the raw spectral data is a good method to continue modelling with, while the abundance data from the CSUM can be used as a benchmark.

The results of the regression analysis on the spectral data show that the gradient boosting algorithm, which is an ensemble model, performs relatively better than the linear, non-parametric and neural models. While on abundance data, linear regression performs relatively better than the non-parametric, ensemble and neural models. Either of the models on either of the data can be employed to obtain the blood response image which can be used to qualitatively sort a salmon fillet as good or bad. However, more work is required to improve the performance of the models.

## 7.2 Recommendations for Further Work

The results of the regression analysis and explanations are presented in section 6.2 followed by section 6.3 in which my personal analysis and interpretation on the steps involved in this thesis are explored in detail. This section explores further into the future works and ideas that can use this thesis as a foundation.

As explained in section 6.3, this thesis can be extended into a binary classification problem by setting a threshold value. But this threshold cannot be a constant value as it depends on each salmon fillet. I would recommend considering setting an average value for the fillets from the same fish while adjusting it on a suitable scale for the other fillets. This must be approved by professionals with much relevant experience in the salmon industry before being put to use.

The novel method for **ridge detection** described in this thesis as part of the pre-processing as presented in section 4.2 involves experimenting with different colour spaces and works well on the skinned fillets as in the data used in this study. While this method can also be employed with minor modifications, on fillets scanned with the skin-side up which is the focus of an ongoing project at Maritech right now.

One major application of the work in this thesis is to generate more labelled data. The process of chemical analysis on the fillet muscles is time consuming and highly laborious in nature, and hence the scarcity of labelled data. The methods employed in this thesis can be employed to generate more labelled data which can then be used in robust deep learning models. Data augmentation that was used to increase the training samples for Breeze as presented in section 6.4.1. This involves splitting the sample locations up into smaller grids and worked well for getting more training data without having to do more chemical analysis. This improved the model performance significantly.



# List of Figures

1.1	An illustration of the electromagnetic spectrum[7]	3
2.1	A hypercube vs RGB image [9]	8
2.2	Operating principle of a hyperspectral camera [10]	9
2.3	An illustration of Spectral Unmixing [14]	14
2.4	Color images (a) and corresponding blood concentration images (b) from four example fillets [6]	22
2.5	Colour image of cod (a) where green dashed line is the manually marked centreline, blue dotted lines indicate the transition between different parts of the cod and the corresponding ridge enhanced image (b). [16]	25
3.1	Outline of the thesis	35
3.2	Expanded view of the steps of Data Collection & Preparation	36
3.3	Expanded view of the steps of Data Pre-processing	38
3.4	Expanded view of the steps of Feature Engineering	39
3.5	Expanded view of the steps of Machine Learning (ML) Modelling	40
3.6	An overall expanded overview of all the steps involved in this thesis	42
4.1	Salmon fillets tagged and ready to be scanned	45
4.2	An illustration of the Hyperspectral imaging setup [24]	46
4.3	Real-time Hyperspectral imaging setup at Nofima	46
4.4	Sample collection by Nofima crew	47
4.5	The sample fluids	48
4.6	The absorbance values of the salmon pigments (a) and blood (b)	48
4.7	Calibration curve	49
4.8	RGB image of fillet sample 177	50
4.9	Locations of Samples obtained for chemical analysis	51
4.10	The segmented image of fillet 177 before and after rotation.	52
4.11	Segmented and rotated image after bounding.	53
4.12	Applying side detection on fillet 177	54
4.13	Ridge detected image using Skjelvareid et al. method.	54
4.14	Applying step 3 on the image of fillet 177.	55
4.15	Computer centres of the sampling locations.	56
4.16	Computed sampling locations.	57
6.1	Plot of the linear regression models on the spectral framework	72
6.2	Plot of the linear regression models on the abundance framework	73
6.3	Histogram of residual error distributions from Linear Regression.	73
6.4	Histogram of residual error distribution from Ridge Regression.	74

---

6.5	Histogram of residual error distribution from PLS Regression. . . . .	74
6.6	Plot of the non-parametric regression models on the spectral framework .	75
6.7	Plot of the non-parametric regression models on the abundance framework	76
6.8	Histogram of residual error distribution from K-Nearest neighbors. . . . .	76
6.9	Histogram of residual error distribution from Support Vector Machines. .	77
6.10	Plot of the ensemble regression models on the spectral framework . . . . .	78
6.11	Plot of the ensemble regression models on the abundance framework . . .	79
6.12	Histogram of residual error distribution from Random Forest. . . . .	79
6.13	Histogram of residual error distribution from Gradient Boosting. . . . .	80
6.14	Plot of the neural regression model on the spectral framework . . . . .	81
6.15	Plot of the neural regression model on the abundance framework . . . . .	82
6.16	Histogram of residual error distribution from Multi-Layer Perceptron. . .	82
6.17	Plot of the regression models on the spectral framework . . . . .	84
6.18	Plot of the regression models on the abundance framework . . . . .	84
6.19	The Maritech Eye product from the event launch . . . . .	88
6.20	The blood response image from CSUM . . . . .	89
6.21	The RGB image (a) and corresponding blood response image (b) from spectral data . . . . .	91
6.22	The noise in absorbance spectra . . . . .	92

# List of Tables

5.1	The best parameters from the Grid Search for all the ML regression models	66
6.1	Scores and Errors from each Linear model for spectral data . . . . .	74
6.2	Scores and Errors from each Linear model for abundance data . . . . .	74
6.3	Scores and Errors from each Non-parametric model for spectral data . . .	77
6.4	Scores and Errors from each Non-parametric model for abundance data .	77
6.5	Scores and Errors from each Ensemble model for spectral data . . . . .	80
6.6	Scores and Errors from each Non-parametric model for abundance data .	80
6.7	Scores and Errors from each Neural model for spectral data . . . . .	82
6.8	Scores and Errors from each Non-parametric model for abundance data .	83
6.9	Scores and Errors from each models for spectral data . . . . .	85
6.10	Scores and Errors from each models for abundance data . . . . .	85





# Appendix A

## Salmon Health Certificate

### Product certificate



**Order number:** 0011559807  
**PO number:** KURT OPPEDAL  
**Harvest ID:** 190447

#### Product information

Fish group:	NSMVa219AM60-SBuk192	Species:	Atlantic Salmon
Generation:	2019	Egg strain:	Mowi
Transfer period:	Q3 : Jul-Sep	Smolt type:	0+

#### Marine production site

Farm name:	Buksevika	Cage no:	0005
Site code:	BAA	Max Density:	25 kg/m <sup>3</sup>
Location no:	11857		

#### History of production

Egg supplier:	Tveitevåg	Waiting cage:	
Smolt supplier:	Vågafossen	Well boat name:	Taupo
Hatching date:	2019-02-06	Harvest date:	2021-02-16
S. water entrance:	2019-10-12	Pack date:	2021-02-17
Last feeding date:	2021-02-08	Primary Packing Station:	Ryfisk R110
Sea temperature:	6	Stunning method:	

#### Fat and colour - product

Date of analysis:	2021-02-01	Fat (%):	14.88
Color (Salmonfan):	26.20	Condition fact:	1.33
Pigment (mg/kg):	40		

#### Feed last 6 months

To date	Feed supplier	Type of feed	Type of pigment	Pigment (mg/kg)
2021-02-08	Mowi Feed	JUPITER 4000		40

#### Vaccination

Product	Vaccination date
Alpha Dip ERM Salar	2019-08-23
Alphaject Micro 6	2019-08-23

#### Treatments

Product	Last day of treatment	Withdrawal time	Time from last treatment to harvest	Avg temp last treatment to harvest	First allowed harvest date
Aquacen 380 mg/ml	2018-12-31	500 daydegrees	More than one year	10	2019-02-27
Slice 20mg/kg 500ddg	2020-03-30	500 daydegrees	3721 ddg	12	2020-05-12
Optilicer	2020-08-07	0 none	193 days	12	2020-08-07
Finquel 100% 21 days	2020-08-28	21 days	172 days	11	2020-09-18
Thermolicer	2020-09-19	0 none	150 days	11	2020-09-19
Benzoak 200mg/ml	2020-09-25	7 daydegrees	2165 ddg	10	2020-09-26
Benzorion	2020-12-29	7 daydegrees	367 ddg	6	2020-12-30
Optomease	2021-02-09	7 daydegrees	38 ddg	6	2021-02-10

# Bibliography

- [1] Life in Norway. Norway's Biggest Industries. Available at [link](#) (2019/02/10).
- [2] Norwegian Seafood Council. 2020. Norwegian seafood exports top NOK 107 billion in 2019. Available at [link](#) (2020/01/07).
- [3] T. Sokolova. Determination of quality metabolites in different brands of salmon fillets. Master's thesis, Trondheim, 2017.
- [4] E. Misimi U. Erikson. "atlantic salmon skin and fillet color changes effected by perimortem handling stress, rigor mortis, and ice storage". *Journal of food science*, 73:C50–C59, 2013. doi: 10.1111/j.1750-3841.2007.00617.x.
- [5] Stein Harris Olsen, Nils Kristian Sorensen, Svein Kristian Stormo, and Edel Oddny Elvevoll. Effect of slaughter methods on blood spotting and residual blood in fillets of Atlantic salmon (*Salmo salar*). *Aquaculture*, 258(1):462–469, 2006. ISSN 0044-8486. doi: <https://doi.org/10.1016/j.aquaculture.2006.04.047>.
- [6] Martin H. Skjelvareid, Karsten Heia, Stein Harris Olsen, and Svein Kristian Stormo. "Detection of blood in fish muscle by constrained spectral unmixing of hyperspectral images". *Journal of Food Engineering*, 212:252–261, 2017. ISSN 0260-8774. doi: <https://doi.org/10.1016/j.jfoodeng.2017.05.029>.
- [7] Cyberphysics. The electromagnetic spectrum: the family of light. Available at [link](#) (2021/05/21). Online; accessed 25 May 2021.
- [8] Silja Bogfjellmo. Hyperspectral analysis of plastic particles in the ocean, 2016.
- [9] RGB. Hyperspectral Image Reconstruction from RGB Image. Available at [link](#) (2020/06/17).
- [10] Norsk Elektro Optikk. What is hsi.
- [11] J D Ingle, Jr and S R Crouch. *Spectrochemical analysis*.
- [12] National Tsing Hua University. Beer-Lambert Law.

- [13] Peg Shippert. Introduction to hyperspectral image analysis. *Online Journal of Space Communication*, 01 2003.
- [14] Spectral unmixing. An overview on hyperspectral unmixing. Available at [link](#).
- [15] Mario Parente and Antonio Plaza. Survey of geometric and statistical unmixing algorithms for hyperspectral images. In *2010 2nd Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing*, pages 1–4, 2010. doi: 10.1109/WHISPERS.2010.5594929.
- [16] Agnar Holten Sivertsen, Chih-Kang Chu, Lih-Chung Wang, Fred Godtliebsen, Karsten Heia, and Heidi Nilsen. Ridge detection with application to automatic fish fillet inspection. *Journal of Food Engineering*, 90(3):317–324, 2009. ISSN 0260-8774. doi: <https://doi.org/10.1016/j.jfoodeng.2008.06.035>.
- [17] Angelo Sassaroli and Sergio Fantini. Comment on the modified Beer–Lambert law for scattering media. *Physics in Medicine and Biology*, 49(14):N255–N257, jul 2004. doi: 10.1088/0031-9155/49/14/n07. URL <https://doi.org/10.1088/0031-9155/49/14/n07>.
- [18] D T Delpy, M Cope, P van der Zee, S Arridge, S Wray, and J Wyatt. Estimation of optical pathlength through tissue from direct time of flight measurement. *Physics in Medicine and Biology*, 33(12):1433–1442, dec 1988. doi: 10.1088/0031-9155/33/12/008. URL <https://doi.org/10.1088/0031-9155/33/12/008>.
- [19] Nils Kristian Afseth and Achim Kohler. Extended multiplicative signal correction in vibrational spectroscopy, a tutorial. *Chemometrics and Intelligent Laboratory Systems*, 117:92–99, 2012. ISSN 0169-7439. doi: <https://doi.org/10.1016/j.chemolab.2012.03.004>. URL <https://www.sciencedirect.com/science/article/pii/S0169743912000494>. Special Issue Section: Selected Papers from the 1st African-European Conference on Chemometrics, Rabat, Morocco, September 2010 Special Issue Section: Preprocessing methods Special Issue Section: Spectroscopic imaging.
- [20] Nirmal Keshava. A survey of spectral unmixing algorithms. *Lincoln Laboratory Journal*, 14:55–78, 01 2003.
- [21] G.H. Golub and Victor Pereyra. The differentiation of pseudo-inverses and nonlinear least squares problems whose variables separate. *SIAM Journal on Numerical Analysis*, 10:413–432, 04 1973. doi: 10.1137/0710036.
- [22] V.G. Doví, E.P. Arato, and L. Maga. A more general formulation of separable least squares. *Mathematical Modelling*, 8:20–23, 1987. ISSN 0270-0255. doi: [https://doi.org/10.1016/0270-0255\(87\)90001-1](https://doi.org/10.1016/0270-0255(87)90001-1).



---

//doi.org/10.1016/0270-0255(87)90533-1. URL <https://www.sciencedirect.com/science/article/pii/0270025587905331>.

- [23] Rasmus Bro and Sijmen De Jong. A fast non-negativity-constrained least squares algorithm. *Journal of Chemometrics*, 11(5):393–401, 1997. doi: 10.1002/(SICI)1099-128X(199709/10)11:5<393::AID-CEM483>3.0.CO;2-L.
- [24] K. Heia, K. E. Washburn, and M. H. Skjelvareid. Automatic quality control of internal defects in cod - results from hyperspectral, ultrasound and x-ray imaging. Technical report, 2017.