

PALACKÝ UNIVERSITY OLMOUC  
FACULTY OF SCIENCE  
DEPARTMENT OF OPTICS

**MASTER'S THESIS**

Multispectral Image Analysis Using  
Artificial Neural Networks



Author:	<b>Ondřej Vaculík</b>
Programme of study:	N1701 Physics
Field of study:	1701T029 Optics and Optoelectronics
Study mode:	Full-time
Supervisor:	Ing. Zdeněk Řehoř Ph.D.
Thesis submitted in:	June 2020

UNIVERZITA PALACKÉHO V OLMOUCI  
PŘÍRODOVĚDECKÁ FAKULTA  
KATEDRA OPTIKY

**DIPLOMOVÁ PRÁCE**

Analýza multispektrálních snímků  
s využitím umělých neuronových sítí



Vypracoval:	<b>Ondřej Vaculík</b>
Studijní program:	N1701 Fyzika
Studijní obor:	1701T029 Optika a optoelektronika
Forma studia:	Prezenční
Vedoucí:	Ing. Zdeněk Řehoř Ph.D.
Odevzdáno:	Červen 2020

### **Poděkování**

Zde bych chtěl srdečně poděkovat svému vedoucímu Ing. Zdeňkovi Řehořovi Ph.D. za cenné rady a konzultace k této závěrečné práci. Rovněž bych chtěl poděkovat RNDr. Jaroslavu Wagnerovi Ph.D. za pomoc s 3D tiskem filtrového kola. Děkuji spolužákům za vzájemnou motivaci. Také bych chtěl poděkovat své rodině, přítelkyni, kamarádům a kapele HVOB za veškerou podporu při studiu i mimo něj.

### **Čestné prohlášení**

Prohlašuji, že jsem předloženou diplomovou práci vypracoval samostatně pod vedením Ing. Zdeňka Řehoře Ph.D. a že jsem použil pramenů, které cituji a uvádím v seznamu použitých zdrojů. Souhlasím, aby byla práce použita pro potřeby výuky a prezentována na internetových stránkách katedry optiky.

V Olomouci dne .....

.....  
Ondřej Vaculík

## Bibliografická identifikace

Jméno a příjmení autora	Ondřej Vaculík
Název práce	Analýza multispektrálních snímků s využitím umělých neuronových sítí
Typ práce	Diplomová
Pracoviště	Katedra optiky
Vedoucí práce	Ing. Zdeněk Řehoř Ph.D.
Rok obhajoby práce	2020
Abstrakt	Náplní závěrečné práce je klasifikace spektrálního průběhu jednotlivých obrazových bodů v pořízených multispektrálních (MS) snímcích zvolené scény – travnatého porostu a stromů v různých částech roku – pomocí metod strojového učení a umělých neuronových sítí. Teoretická část práce uvádí základy vizualizace MS snímků, metod strojového učení, problematiku segmentace vícerozměrných datových sad a příklady postupů a algoritmů vhodných pro tuto úlohu. V experimentální části je provedena implementace těchto algoritmů na pořízených MS snímcích. Získané třídy objektů jsou porovnány se spektrální databází. Tyto výstupy poté slouží pro zhodnocení zdravotního stavu snímané zeleně.
Klíčová slova	CCD, multispektrální snímání, segmentace obrazu, autoencoder, shlukování, neuronová síť, strojové učení, dálkový průzkum
Počet stran	48
Počet příloh	7
Jazyk	Anglický

# Bibliographical identification

Author's first name and surname	Ondřej Vaculík
Title	Multispectral Image Analysis Using Artificial Neural Networks
Type of thesis	Master's
Department	Department of Optics
Supervisor	Ing. Zdeněk Řehoř Ph.D.
The year of presentation	2020
Abstract	The scope of the thesis is the classification of the spectral response of individual pixels acquired by a multispectral (MS) imaging. A selected scene – lawn with grass and trees observed in different parts of the year – is analysed using machine learning and neural network algorithms. The theoretical part of the work presents the basics of MS image visualisation, machine learning methods, the problem of segmentation of multidimensional data sets and examples of procedures and algorithms suitable for this task. In the experimental part, the implementation of these algorithms is performed on acquired MS images. Obtained object classes are compared with a spectral database. These outputs are then used to evaluate the vitality of inspected plants.
Keywords	CCD, multispectral imaging, image segmentation, autoencoder, clustering, neural network, machine learning, remote sensing
Number of pages	48
Number of appendices	7
Language	English

# Contents

<b>Introduction</b>	<b>7</b>
<b>1 Theoretical Part</b>	<b>8</b>
1.1 Motivation . . . . .	8
1.2 Multispectral Imaging . . . . .	8
1.3 Multispectral Data Processing . . . . .	13
1.4 Data Processing Example . . . . .	21
1.5 Theoretical Part Conclusion . . . . .	26
<b>2 Experimental Part</b>	<b>27</b>
2.1 Overview . . . . .	27
2.2 Multispectral Image Acquisition . . . . .	27
2.3 Multispectral Image Analysis . . . . .	32
2.4 Time Series Comparison . . . . .	40
2.5 Experimental Part Conclusion . . . . .	42
<b>Conclusion</b>	<b>44</b>
<b>List of Abbreviations and Symbols</b>	<b>45</b>
<b>Bibliography</b>	<b>46</b>
<b>Contents of Enclosed DVD</b>	<b>48</b>
<b>Appendices</b>	

# Introduction

Spectral imaging is a non-destructive method that allows acquisition of both spectral and spatial information related to the observed object. The image is composed of multiple bands capturing selected parts of the electromagnetic spectrum. Several techniques allow this type of imaging, one of which takes the advantage of a scientific CCD camera combined with narrow-band spectral filters. Such setup allows imaging in the range from near-ultraviolet to near-infrared part of the spectrum. With a low number of bands, the output image is called multispectral (MS). As a result of the high dimensionality of the MS image, it is difficult to visualise and interpret the MS image without additional processing.

In this work, we will focus on MS image analysis using conventional remote sensing indices as well as advanced machine learning and neural network algorithms. The theoretical part will focus on the basics of MS image acquisition and visualisation, machine learning methods of multidimensional data analysis and neural network designs aimed towards data representation. Selected approaches will be showcased on an example dataset. The experimental part will describe the MS image acquisition for a selected scene – lawn with grass and trees observed in different parts of the year. We will test the proposed methods on the acquired dataset with the goal of evaluating the change in plant vitality throughout the year. The identified spectral response of the observed objects will also be compared to a reference spectral library.

# Chapter 1

## Theoretical Part

### 1.1 Motivation

Digital photography enables a fast and convenient way of capturing images, that can then be analysed using various image processing methods. The nature of the captured colour image is optimised to be easily interpreted by human vision. A conventional digital image is composed of individual picture elements (pixels). Each pixel consists of three components representing colour – red, green and blue (RGB). The algorithms utilising colour images as input are aimed towards feature extraction, object or pattern recognition – tasks similar to those performed by the visual cortex. With the rising popularity of artificial neural networks and machine learning, more robust solutions to these problems are being developed.

Inevitably, the information about the colour of the object represented by the RGB values is in some cases insufficient. Assume one wants to identify a chemical composition or other properties of the observed object. This task can be hard or even unsolvable using RGB image as an input. Here the method of spectral imaging can be used to obtain a more detailed spectral response from the observed materials. In combination with neural network and other algorithms, the chance to solve such problem increases.

### 1.2 Multispectral Imaging

Spectral imaging is a specialised method of imaging where one obtains images corresponding to a number of spectral channels. The used spectral regions are often at least partially outside the visible spectral range, covering parts of the infrared and ultraviolet region [1]. Each pixel of the final image contains spectroscopic information defined by the used spectral channels.

The main two categories of spectral imaging are multispectral (MS) and hyperspectral (HS) imaging. The main difference between the two is the number of detected spectral bands and their coverage of the detected spectral region. MS images contain up to 10 bands and each band is several nanometers wide. For HS images the number of bands can reach hundreds or thousands and each band tends to be very narrow, usually 1 nm or less.

For some tasks, MS imaging provides a fair trade-off between the amount of retrieved information versus the complexity of the sensing instrument. Figure 1.1 presents the main difference between MS imaging when compared to colour imaging utilising the standard RGB detection in terms of detection bands.



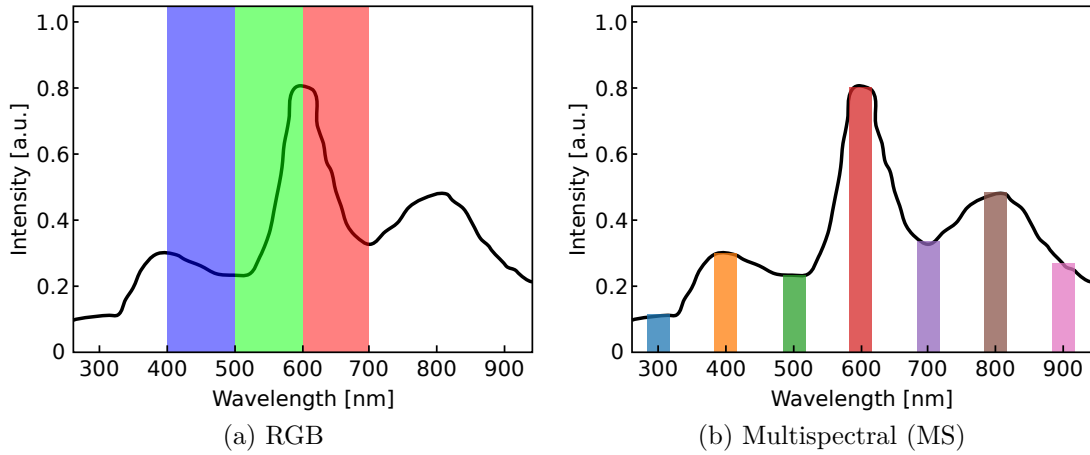


Figure 1.1: Comparison of detection bands for an arbitrary spectral response.

As the comparison suggests, MS imaging provides a more accurate representation of the reflectance spectra than RGB detection. On the other hand, when compared with spectroscopy, MS imaging yields a response with a lower spectral resolution, but it combines it with spatial information, which is not measured using standard spectroscopic methods. This can be exploited in image analysis, where the spectral information can be used to identify materials inside the scene. Multispectral instruments are often mounted on satellites or aeroplanes for remote sensing applications. They can also be used in medicine, agriculture or authentication purposes, as explored in [2].

### 1.2.1 Multispectral imaging using CCD camera

Several methods that allow capturing such images exist, ranging from simple setup of charge-coupled device (CCD) chip paired with bandpass spectral filters to complex systems of optical gratings or detectors sensitive to particular wavelengths. This thesis focuses on the first approach. By utilising a scientific camera with CCD chip a range from near-ultraviolet to near-infrared part of the electromagnetic spectrum can be detected (Fig. 1.2).

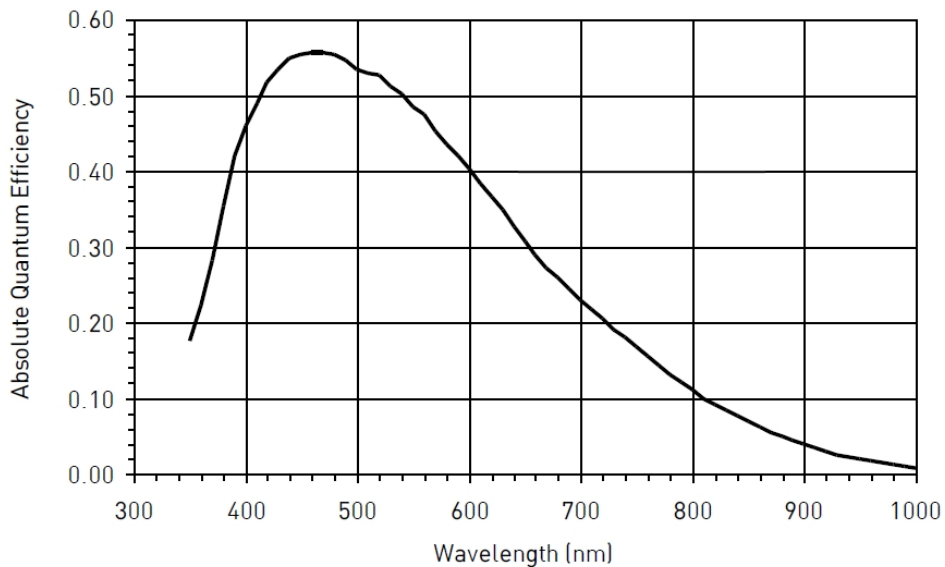


Figure 1.2: Quantum efficiency of CCD chip KAI-4022 [3].

The main advantage of a scientific camera is the ability to capture images with high signal to noise ratio (SNR). This is possible thanks to a large sensor area and pixel size combined with suitable cooling to lower the thermal noise. This is particularly important when using a camera together with narrow-band spectral filters. The number of photons reaching the sensor area is lower, therefore the image exposure time must be extended. This is also combined with image calibration techniques – dark image subtraction, flat field correction and image averaging. All of them are used to improve the SNR of the final image and to reduce the effect of the lens aberrations and detector noise.

Dark image subtraction is a technique for reduction of the thermal noise present on the CCD chip. Random detection occurs due to presence of thermal electrons and their random excitation – they add noise to the image. The rate of this random detection can be reduced by cooling the CCD chip, but its total elimination is not possible. Therefore it is useful to acquire dark exposures, taken while the chip is being obstructed from external light by cover or mechanical shutter. The dark exposure image reflects the random thermal noise present solely due to thermal fluctuations. It is also important to set the exposure time of the dark image same as for the image containing the measured object (light image) because the amount of noise varies with the length of exposure as well. The dark image represents the amount of thermal noise added to the image. This dark exposure can be then subtracted from the light image to lessen the effect of the additive noise.

Flat field correction aims to reduce the multiplicative noise and distortion introduced by the optical elements. The lens aberrations and vignette effects are independent of temperature but are unique for each lens and filter used in light image capture. Flat field image is taken with the optics used for light image acquisition, but the lens is aimed at an evenly lit surface with a neutral colour (white or grey). The flat field image holds information about vignetting and uneven light distribution. The correction of these aberrations is achieved by dividing the light image by the flat field image.

Image averaging is a simple method of reducing the random reading and electronic noise present in the light image. Multiple exposures of the scene can be taken and averaged out to reduce the uncertainty in the measured data. This also applies for the dark and flat field images, as their respective averages provide a more accurate representation of the noise they aim to reduce. These noise reduction methods with respect to CCD imaging are thoroughly discussed in [4].

The MS image acquisition using a CCD camera with external filters described step by step:

- Mount camera on a tripod and connect the filter carousel.
- Connect to PC and set cooling temperature for the CCD chip.
- Adjust the composition and focus.
- For every filter in the filter carousel:
  - Place narrow-band filter in front of the lens.
  - Set appropriate exposure to avoid under or overexposure.
  - Acquire multiple exposures with the same filter (e.g. 10 for each filter).
- Acquire multiple (e.g. 10) dark frames for the set CCD chip temperatures and each light image exposure time.
- Acquire multiple (e.g. 10) flat field images using the lens and filter.

At this point, two main sets of images are obtained: light images – capturing the measured object, and calibration images – dark and flat field. To perform the image calibration, the images are processed as follows:

- Calculate the average of each dark image exposure time.
- Calculate average of flat field images.
- Calculate the average of each light image (per used filter).
- Subtract dark image mean from the light image mean.
- Divide the light image mean by the flat field mean.

Now the number of images at hand matches the number of used filters. These images require further processing to form the final MS image. The difference in spectral sensitivity of the CCD chip for different wavelengths must be taken into account (Fig. 1.2). The spectral composition of the light illuminating the scene affects the reflectance as well. Solar radiation serves as the main source of illumination for exterior imaging, which is the case for our measurement. Its spectral composition after passing through the Earth's atmosphere is plotted in Figure 1.3.

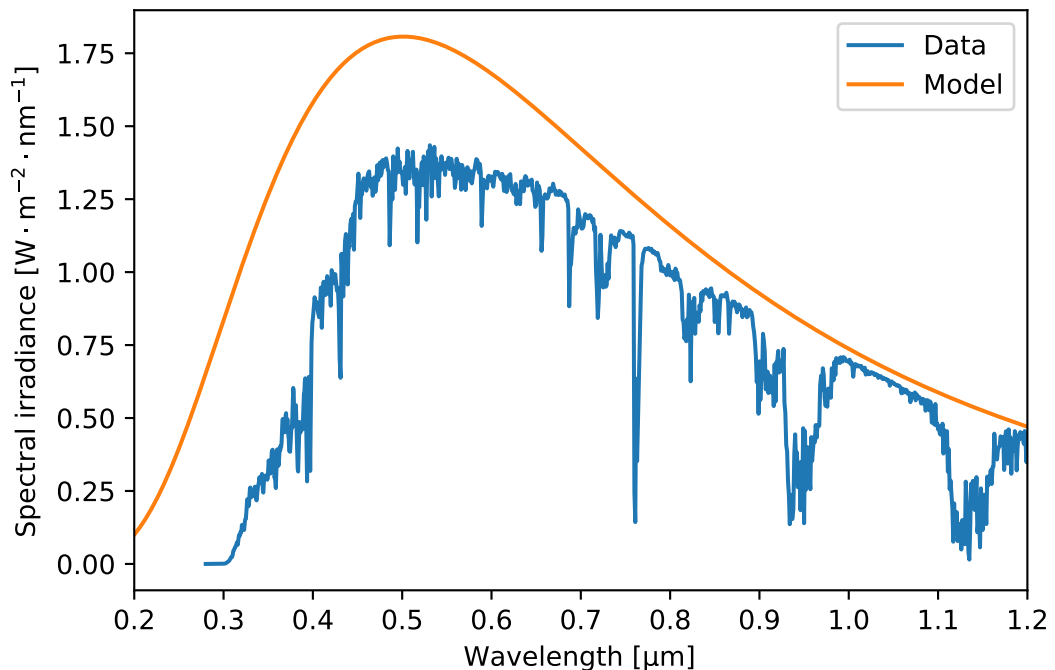


Figure 1.3: Solar spectral irradiance of the Earth surface (blue) with blackbody emission curve for  $T = 5780$  K (orange) [5].

The function describing the total spectral response becomes non-trivial, due to its dependence on the properties of the illumination and detector. Image normalisation is used to compensate for these effects. Histogram stretching is performed as a first step in the normalisation process. Pixel values are adjusted to match a predefined range to increase the dynamic range. Histogram stretching is accomplished using equation 1.1.

$$I_E = (I - \text{Min}) \times \frac{\text{newMax} - \text{newMin}}{\text{Max} - \text{Min}} + \text{newMin} \quad (1.1)$$

$I_E$  is pixel value after stretching,  $I$  is input pixel value, **Min** (**Max**) represents pixel value of the input image and **newMin** (**newMax**) is the target pixel value after histogram stretching. The target values are chosen inside the available value range (bit depth) to avoid clipping in the following step, which includes manipulation of the values of the entire image.

After image histogram stretching, image normalisation is performed. A sufficiently large white sheet of paper is used as a normalisation reference. The first step is to calculate the mean pixel value  $m_i$  of the white paper area for each image. From each individual mean the main mean  $M$  is calculated, which serves as a reference point. After that, all pixel values of each image are adjusted so that the white normalisation area has a same mean value as the main mean (Eq. 1.2).

$$I_N = I_E + (M - m_i), \quad (1.2)$$

$I_N$  is the pixel value after normalisation. This step concludes the image normalisation process. The obtained images form the final MS image. To sum up, the entire process of MS imaging using a CCD camera, the steps are:

1. For each filter acquire multiple light exposures.
2. Acquire corresponding sets of dark and flat field images.
3. Average respective exposures.
4. Calibrate the light images using dark and flat field images.
5. Perform histogram stretching to increase the dynamic range.
6. Adjust the image values to normalise the data using reference surface.

Advantages of this setup are low price compared to the specialised MS cameras and the variability in number and type of used filters. The spatial resolution is also high, thanks to the large sensor and cooling capabilities of the scientific camera. The main disadvantage is the need for long exposure times, therefore it can be only used for imaging static objects. Using external filters also reduces the portability of the whole setup and does not allow eg. mounting it on a drone. The output of this setup after the calibration and normalisation process are individual images that form the final MS image.

### 1.2.2 Multispectral image visualisation

After an MS image is captured, one might try to evaluate the contents by simple observation. A problem arises – how to visualise the MS image? The first option is to view each band as a separate grayscale image, but the information between bands is therefore omitted. The second option is to use the false-colour technique. Selected bands can be mapped to colour channels to compose a false-colour RGB image. In remote sensing, RGB can be mapped to near-infrared (NIR), red and green to highlight vegetation in a red tone, as it reflects most light in the NIR part of the spectrum (Fig. 1.4).

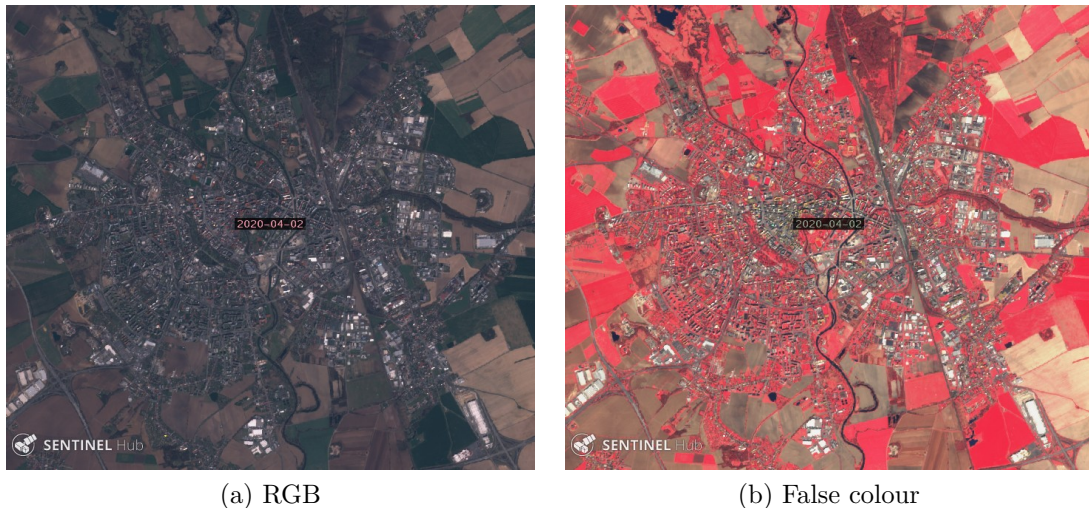


Figure 1.4: Satellite image of Olomouc, taken by Sentinel-2 L1C [6].

When using false colour, part of the information is still unused because the total number of bands in MS image usually exceeds the number of colour channels available in RGB colour space. To take advantage of the information contained in the whole MS image fully, more complex methods are required. The following chapter describes a few approaches to more complex MS image processing.

## 1.3 Multispectral Data Processing

The information richness of the MS data is higher than conventional colour images. For true colour RGB image with 8-bit depth per colour channel, the total possible number of combinations is  $2^{24}$ . Sensing instruments for MS images usually exceed the depth of 8 bits per channel. For a 6 band MS image with a depth of 16 bits per band gives a total of  $2^{96}$  possible combinations. With this also comes an additional complexity to present and interpret the image contents. Additional analysis is needed to extract the underlying information.

Firstly let us take a look at remote sensing indices, arithmetic operations done on selected bands to highlight certain features. Then the possibility to look at the data as  $n$ -dimensional space (sometimes referred to as data cube or spectral cube) is explored, where  $n$  denotes the number of bands. This is followed by a  $k$ -means clustering, used to find similarities in data. Finally, a neural network is trained for data compression – autoencoder – to visualise  $n$ -dimensional dataset in a lower dimension. The combination of listed methods can be used to achieve the best results for individual output requirements.

### 1.3.1 Remote sensing indices

One of the approaches to MS image analysis other than using false colour is band arithmetic. Specific formulas are derived based on available detection bands and desired output. The output of the formula is called remote sensing index, or just index. The main area of use is MS satellite image analysis. Some of the indices are tailor-made to suit specific sensor equipment on-board the satellites or to compensate for the effects

of the atmosphere. This section introduces two indices used for vegetation sensing and chlorophyll detection.

### NDVI – Normalised Difference Vegetation Index

This index is one of the most profound remote sensing indices used for green plant assessment. The NDVI reflects the presence of vegetation derived from the photosynthetic capacity of the green canopy [7]. The NDVI formula is stated in equation (1.3)

$$\text{NDVI} = \frac{\text{NIR} - \text{R}}{\text{NIR} + \text{R}}, \quad (1.3)$$

where NIR stands for near-infrared and R for red band spectral reflectance. The NDVI index varies in range  $\langle -1,1 \rangle$ . Areas of the image containing dense vegetation tend to positive values, in contrast with e.g. water, which produces low positive or negative NDVI values.

### CI-G – Chlorophyll Index Green

Chlorophyll index green is constructed as a leaf chlorophyll estimator. Its formula, presented in equation (Fig. 1.4), is based on the spectral properties of chlorophyll. Higher values indicate higher concentrations of chlorophyll.

$$\text{CI-G} = \frac{\text{NIR}}{\text{G}} - 1 \quad (1.4)$$

The CI-G index utilises near-infrared (NIR) and green (G) bands. Figure 1.5 shows a spectral absorption of two main photosynthetic pigments – chlorophyll a (CH-a) and chlorophyll b (CH-b). Red wavelengths are strongly absorbed by chlorophyll; as chlorophyll content increases, changes in red reflectance will be smaller per unit change of chlorophyll [8].

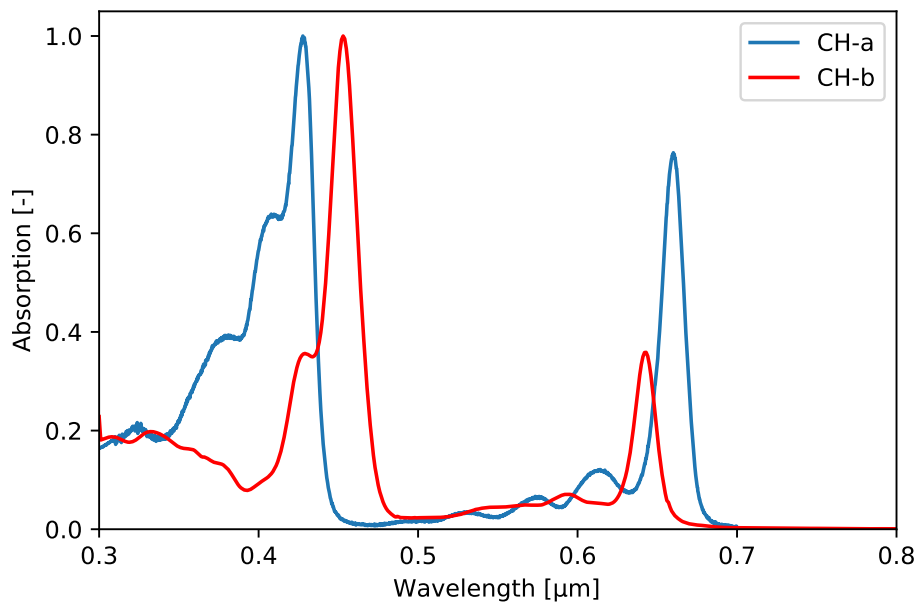


Figure 1.5: Spectral absorption of chlorophyll a and b [9].

### 1.3.2 Machine learning

The next step in MS data processing is to make use of the entire  $n$ -dimensional data cube. The methods of machine learning are more than suitable for this application.

Machine learning (ML) is a scientific study of algorithms which perform tasks without explicit instructions using inference and patterns in input data. ML algorithms build a model using input data (called training set) that makes predictions based on input values. The model parameters are tuned using part of input data (called test set), that is not available to the model during its training stage. The performance of the model can be valued by the accuracy of the predictions made on the test set. ML is closely related to data mining, optimisation, and statistics. The main tasks for ML models include classification, regression, time series analysis, anomaly detection, and many more. There are three main branches of learning types for ML models: supervised, unsupervised and reinforcement learning.

Supervised learning, sometimes defined as learning from example, works with labelled input data – each datapoint consists of input values and a corresponding label. For example, in the task of recognising handwritten digits, the input data can be a grayscale image of digit and the label identifying the number. The performance of the model is measured by comparing the output predicted value with the true value (label) of the trained feature and the model is optimised to minimise this error.

Unsupervised learning is about modelling data that does not contain corresponding labels. The model is given data in an undivided form, simply as a set of examples. The algorithm then tries to generalise the data or find underlying patterns. Two of the most powerful tools for unsupervised ML are clustering and dimensionality reduction.

Reinforcement learning works with models that operate in a simulated environment. The algorithm (called an agent) performs actions to maximise a defined score that directs the agent toward achieving a defined goal. The training data is generated by the algorithm deciding between exploration and exploitation of environment (or state space) and by receiving feedback in the form of the score. The main application of reinforcement learning is in the field of robotics and also time series prediction like stock markets and many others [10].

MS data represent a perfect input for an unsupervised clustering algorithm. The goal here is to find parts of an image that exhibit similar spectral properties. This operation is also called image segmentation. The task of unsupervised data clustering can be resolved using  $k$ -means algorithm, which is described in the following section.

### 1.3.3 The $k$ -means clustering algorithm

The underlying principle of  $k$ -means is to put  $n$  data points in an  $i$  dimensional space into  $k$  clusters. Each cluster is then characterised by a total mean of its components. The data points are treated as vectors and the distance between them is computed using a defined measure function. If the data points live in a real vector space, the distance measure of choice is normal Euclidean distance, defined as follows:

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} = \|\mathbf{x} - \mathbf{y}\|, \quad (1.5)$$

where  $d$  is the normal Euclidean distance and  $\mathbf{x}, \mathbf{y}$  denote vector of two data points. A visual example of  $k$ -means clustering is presented in Figure 1.6.

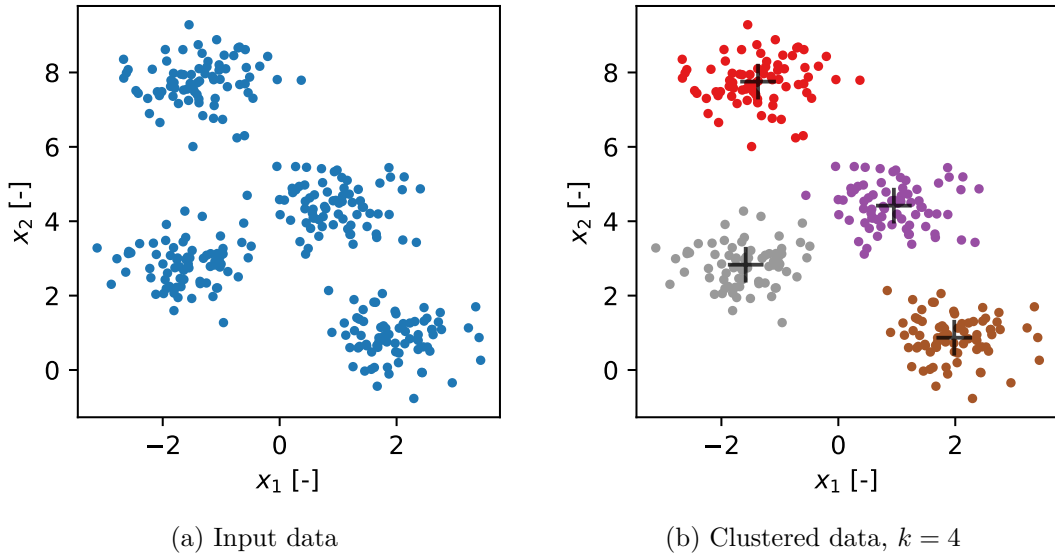


Figure 1.6: Example of  $k$ -means clustering on dummy dataset.

The  $k$ -means algorithm is defined by the following step by step description [11]:

- **Initialisation**

- choose a value for  $k$
- choose  $k$  random positions in the input space
- assign the cluster centres  $\boldsymbol{\mu}_j$  to those positions

- **Learning**

- repeat
  - \* for each datapoint  $\mathbf{x}_i$ :
    - compute the distance to each cluster centre
    - assign the datapoint to the nearest cluster centre with distance

$$d_i = \min_j d(\mathbf{x}_i, \boldsymbol{\mu}_j) \quad (1.6)$$

- \* for each cluster centre:
  - move the position of the centre to the mean of the points in that cluster ( $N_j$  is the number of points in cluster  $j$ ):

$$\boldsymbol{\mu}_j = \frac{1}{N_j} \sum_{i=1}^{N_j} \mathbf{x}_i \quad (1.7)$$

- until the cluster centres stop moving

- **Usage**

- for each test point:
  - \* compute the distance to each cluster centre
  - \* assign the datapoint to the nearest cluster centre with distance as defined by equation (1.6)



One of the questions that arise when using  $k$ -means is how big the number of clusters  $k$  should be? The example provided in Figure 1.6 is visually straightforward, but this can not be applied to higher dimension datasets. One way of doing this is via elbow method – the idea is to run  $k$ -means clustering for a range of clusters  $k$ . Sum of squared distances of each point from its assigned centre is calculated for each  $k$ . This quantity is called distortion  $D$  and is defined in equation 1.8.

$$D = \sum_{i=0}^n \|\mathbf{x}_i - \boldsymbol{\mu}_j\|^2 \quad (1.8)$$

Example of the elbow method is presented in Figure 1.7. The optimal number of clusters for this example is 4 because adding more clusters does not reduce the distortion significantly.

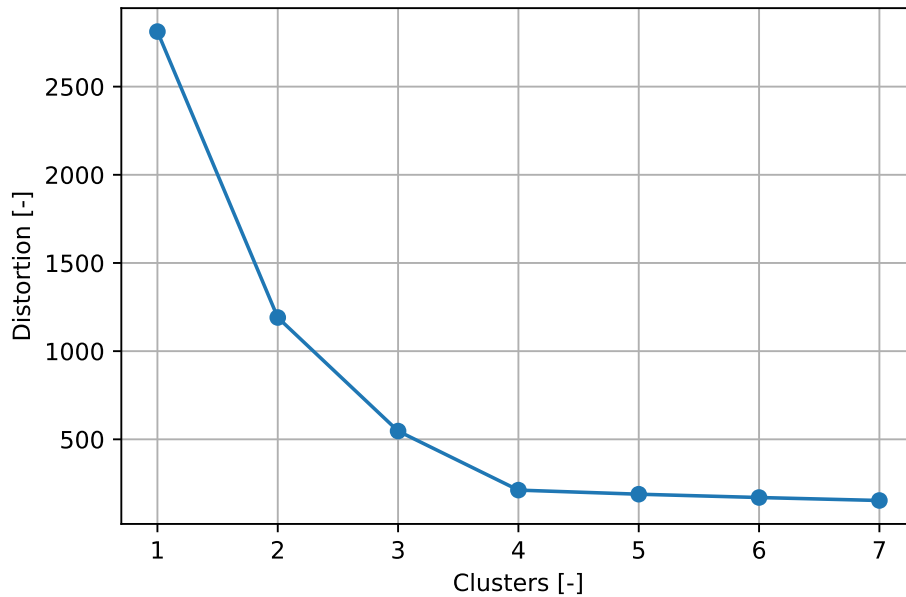


Figure 1.7: Elbow method for dummy dataset from Fig. 1.6.

The  $k$ -means algorithm can as well be used for image compression by colour quantization or supervised classification for input data with labels.

### 1.3.4 Artificial neural network principles

The main building block of a neural network (NN) is a neuron. Following paragraphs define the architecture and the activity rule of a single neuron. A neuron has a number of inputs  $x_i$  and one output  $y$ . Each input is associated with its weight  $w_i$ . There may also be an additional parameter  $b$  called bias, which is not associated with any input. The single neuron works as a feedforward device – it accepts the input values and returns one output value. Last part is an activation function  $f$  [12]. The schematic of a single neuron is pictured in Figure 1.8.

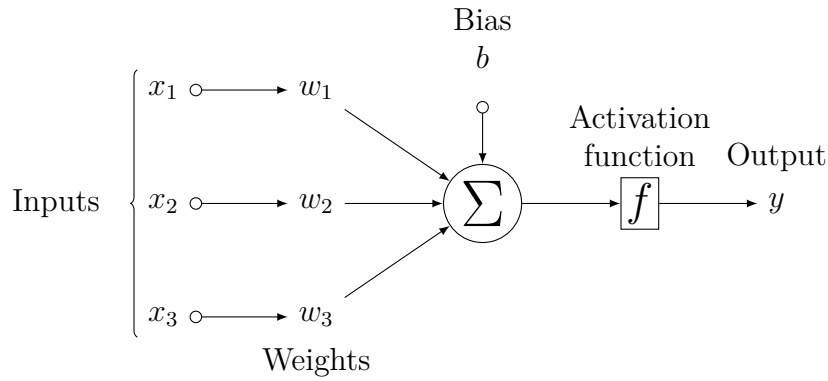


Figure 1.8: Artificial neuron operating principle.

The activity rule of a neuron has two steps: activation  $a$  and output. The activation is simply a product of each input value and corresponding bias all added up:

$$a = \sum_i w_i x_i + b. \quad (1.9)$$

The output is set as a function  $f(a)$  of the activation. The purpose of the activation function is to threshold the input activation and decide the strength of the neurons response. There are several possible activation functions, here are two of the most commonly used:

1. Sigmoid (logistic function)

$$f(a) = \frac{1}{1 + e^{-a}} \quad f \in (0,1) \quad (1.10)$$

2. ReLU (rectified linear unit function)

$$f(a) = \max(0, a) \quad (1.11)$$

A basic NN therefore implements a function  $y(\mathbf{x}; \mathbf{w})$ , where the output  $y$  is a non-linear function of inputs  $\mathbf{x}$ , parameterized by weights  $\mathbf{w}$  and biases  $\mathbf{b}$ .

The main feature of the neuron is the ability to learn, which is accomplished by finding the optimal weights using training. Individual weights combined form a weight space and optimal weight value is determined via error (loss) function during the training process. Example of a supervised learning process is described as following:

1. The weights  $\mathbf{w}$  and biases  $\mathbf{b}$  are set randomly, this is called weight initialisation.
2. Randomly selected input value  $\mathbf{x}^{(j)}$  with label  $t^{(j)}$  is plugged into the network.
3. Output  $y^{(j)}$  of the network is compared with the label  $t^{(j)}$  and error is calculated.
4. The values of the weights are adjusted so that the error becomes smaller.
5. Repeat until the error does not change significantly.

Let's describe the training procedure on a simple example: a binary classifier. Assume a neuron with suitable activation function that has an output between 0 and 1 – called a perceptron. Each of the outputs of such classifier can be interpreted as a probability of belonging to a class. With a first attempt, the input data is fed into

the neuron and it delivers a response. The response can then be compared to the true input label. From this difference, a predefined loss function can be evaluated.

One of the possible ways to define a loss function  $C$  is via mean squared error between the label  $t$  and the prediction  $y$ , defined as:

$$C(\mathbf{w}, \mathbf{b}) = \frac{1}{n} \sum_x \|y(x) - t\|^2, \quad (1.12)$$

where  $n$  is the number of inputs  $\mathbf{x}$  and  $\|\cdot\|$  denotes Euclidean vector norm used as a distance measure. The loss function  $C$  is non-negative and approaches zero if the outputs  $y(x)$  of the NN is close to the target labels  $t$ . Other possible loss functions can be found in [13].

The aim of the training is to minimise the loss function. This could be achieved using optimisation method called gradient descend. The idea is to calculate the gradient of the loss function  $\nabla C$  and change the weights and biases in the direction opposing the calculated gradient to minimise the loss. The step-change  $\Delta \mathbf{v}$  in the NN weights and biases is therefore

$$\Delta \mathbf{v} = -\eta \nabla C, \quad (1.13)$$

where  $\eta$  is a small positive parameter known as the learning rate [14]. There are several methods of gradient calculation or training optimisation like backpropagation, stochastic gradient descend and others. More on this in [13].

If a NN is trained on all available data for a large number of learning rounds (or epochs – pass-through of all the training examples), there is a risk of overfitting. The resulting NN model may perform well on the training data, but results for new unseen data could be completely wrong. The role of a NN engineer is to devise a suitable NN architecture as well as set the appropriate training parameters.

One of the methods to prevent overfitting is to split the data into training and testing sets. The training set is used for training cycle and afterwards, the model is evaluated on the testing set. The 80 to 20 per cent ratio of training to test samples is usually recommended, but this is dependent on the amount and character of the input data as well. This ratio belongs to the set of hyperparameters – values set by the user during the NN design that does not change during the training.

A neuron can be connected with many more adjacent neurons to form a neural network. The design of NN varies with its intended use, mainly the number of input values and desired output. Neurons can also be stacked in layers, the inside neurons are located in the so-called hidden layer. There is no theoretical limit on the number of neurons in input, output or hidden layers, the main limitation is though the computational power to store and manipulate such large amount of data.

When dealing with a large amount of data and not having enough computer memory, batch learning can be used to counter this limitation. The input dataset can be divided into individual batches, each batch having a lower number of samples than the whole dataset. The number of calculations is reduced, instead of calculating the gradient for the whole dataset, the update is done only for a reduced number of samples. Therefore the optimisation can be done on less capable hardware and is achievable with a reasonable time of convergence [11].

### 1.3.5 Dimensionality reduction using autoencoder

Advanced task-specific algorithms can be built using artificial neurons. One of the possible architectures, called autoencoder, is designed for data representation in a lower dimension. The main idea here is to train two separate NN that work together. The first part, the encoder, takes the input values and compresses them into a predefined number of variables. The second part, decoder, takes the compressed representation and reconstructs it back to the original form. The difference between the reconstructed input and original is minimised during the learning process. General schematic of autoencoder is presented below (Fig. 1.9).

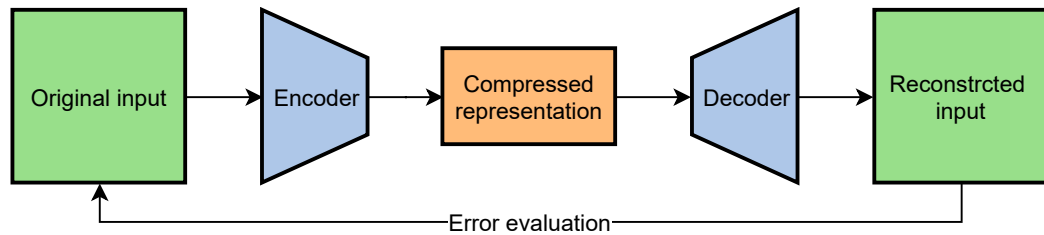


Figure 1.9: Autoencoder principle.

The dimension of the input vector  $\mathbf{x}$  determines the number of the neurons in the input layer of the encoder. The number of variables that the encoder compresses the information into is defined by the NN architecture. The compressed representation vector is labelled  $\mathbf{c}$ .

The compressed representation, also referred to as latent space, serves as a method of denoising as well as data projection technique similar to principal component analysis [11]. The trained autoencoder works as a task-specific lossy data compression algorithm. Compressed representation with two variables is useful as a visualisation tool for high-dimensional data.

## 1.4 Data Processing Example

This section demonstrates the workflow of data compression and processing used in this thesis. The main idea is to perform  $k$ -means clustering on the entire MS image and compare the output with clustering of compressed representation from the autoencoder. For this purpose, an example dataset is being used – MS image of cyan, magenta, yellow and black (CMYK) colour bars printed on a white paper (Appendix 1). The number of classes and their spatial positioning is clearly defined, as well as the spectral response of each colour bar.

Spectral filters utilised in capturing the MS image are listed in Table 1.1. The main parameters describing the filters are  $\lambda_c$  central wavelength of filter,  $\Delta\lambda_c$  uncertainty in the position of central wavelength and FWHM full width in half maximum of the transmittance.

Table 1.1: List of used spectral filters and their properties for CMYK dataset [15].

Filter name	$\lambda_c$ [nm]	$\Delta\lambda_c$ [nm]	FWHM [nm]
IF 450	450	5	$10 \pm 2$
IF 500	500	5	$9 \pm 2$
IF 650	650	7	$9 \pm 2$
IF 750	750	8	$10 \pm 2$
IF 850	850	9	$13 \pm 3$
IF 1000	1000	10	22

Each band can be visualised as a grayscale image to examine the input data before processing (Fig. 1.10).

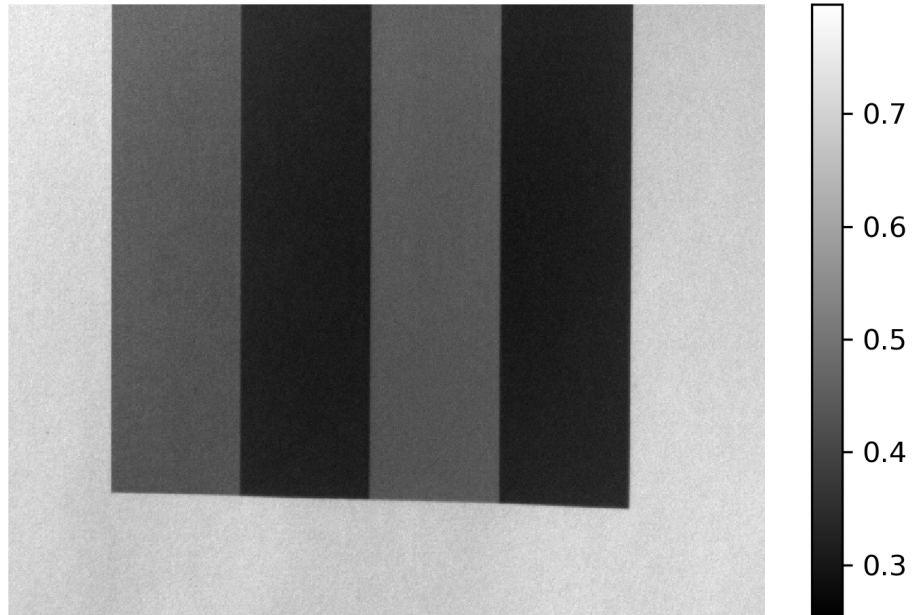


Figure 1.10: One of the MS image layers, taken with IF 500 filter.

The input MS image is transformed to a set of 6-dimensional vectors, that serve as an input for the autoencoder NN training (see Section 1.3.5). The autoencoder represents six input values  $x_i$  with two ( $\mathbf{c} = [c_1, c_2]$ ) in the latent space (Fig. 1.11).

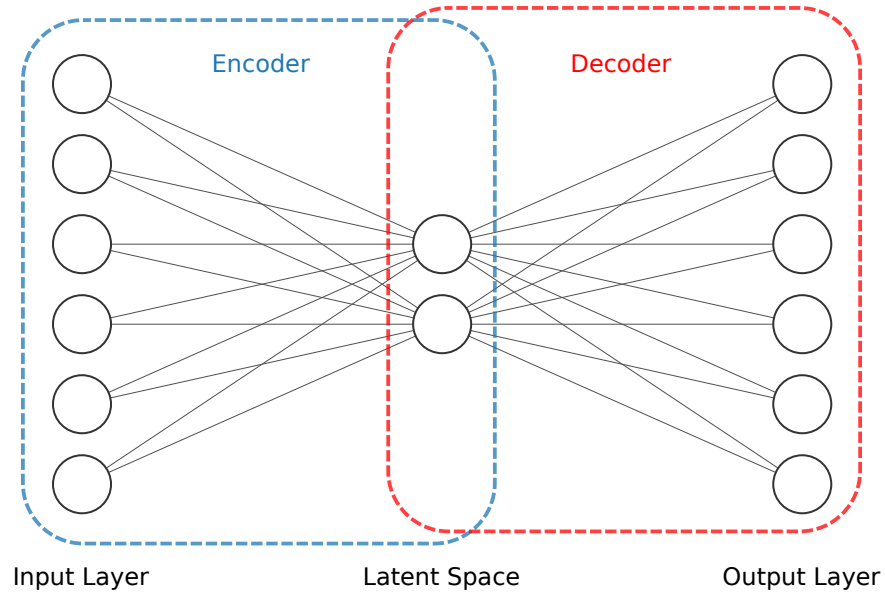


Figure 1.11: Autoencoder neural network design.

The data is randomly split to 80% for training and 20% for testing. Activation function is a sigmoid function (Eq. 1.10). Total number of 20 training rounds (epochs) with a batch size of 1024 were chosen as training parameters. The used optimiser is Stochastic Gradient Descend (SGD) with mean squared error (MSE) as a loss function. The training progress is evaluated by monitoring the loss function for each epoch, calculated on the training and testing set (Fig. 1.12).

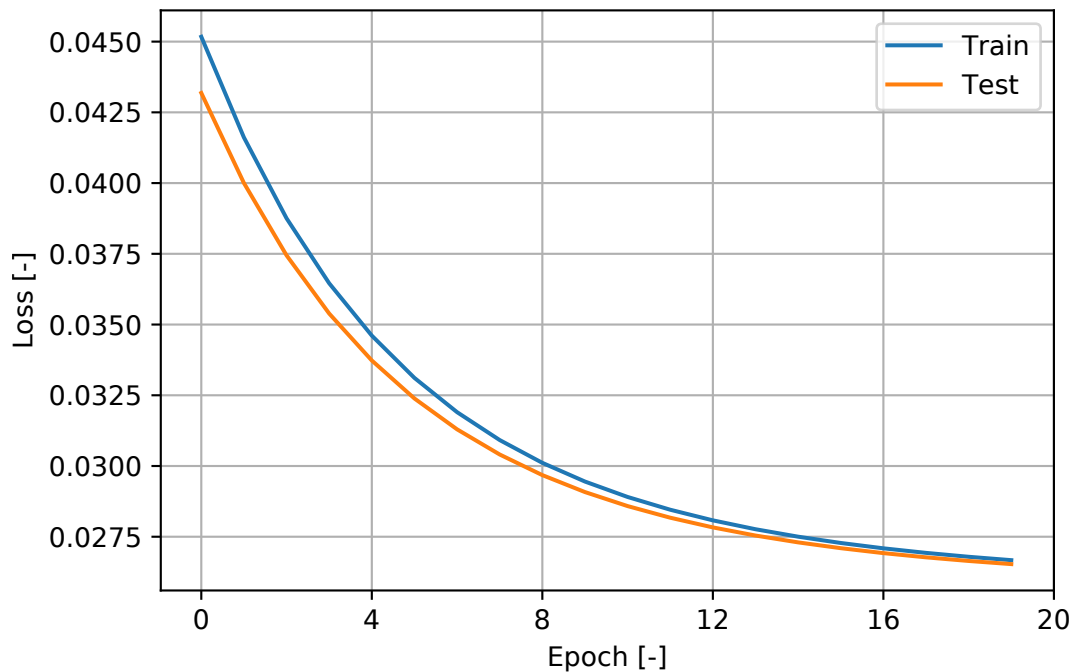


Figure 1.12: AE training loss progress on CMYK data.

The model loss after 20 epochs is very low – the representation of the input data by the model is close to the original values. The resulting compressed representation was used as an input for  $k$ -means algorithm with  $k = 6$ . The uneven illumination of the white background was identified as an additional class. In order to separate the colour bars, the number of clusters has to be increased. The compressed representation of the MS image can be easily plotted along with the classes defined by the  $k$ -means algorithm (Fig. 1.13).

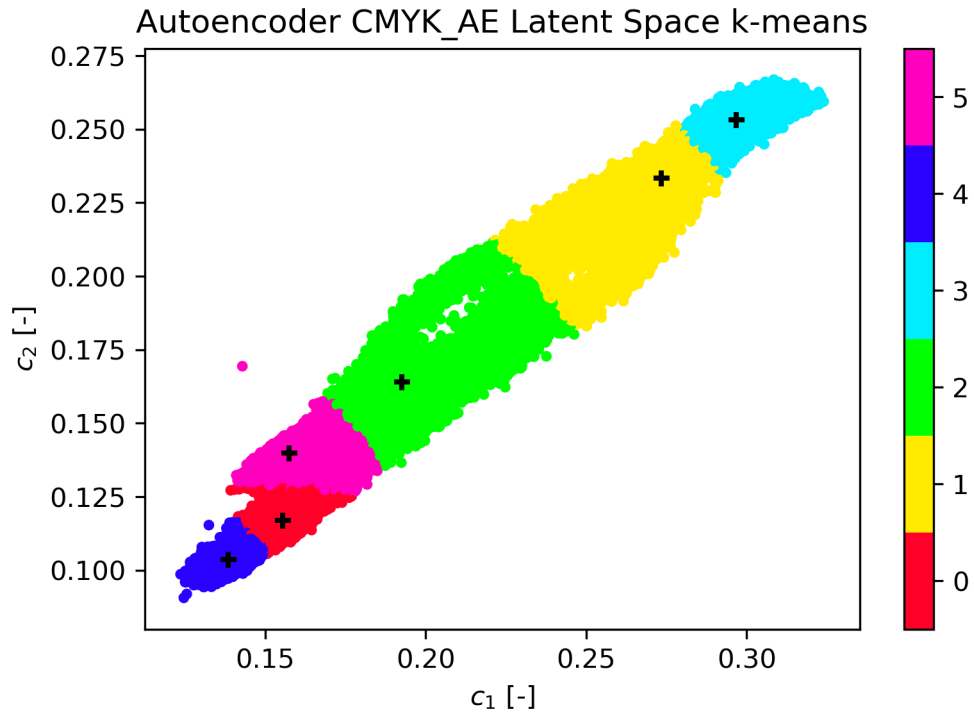


Figure 1.13:  $k$ -means clustering on CMYK data visualised in latent space.

The  $k$ -means algorithm has been evaluated on the entire MS image with the same settings, leaving  $k = 6$ . The identified clusters inside the original image for both approaches can now be visualised (Fig. 1.14).

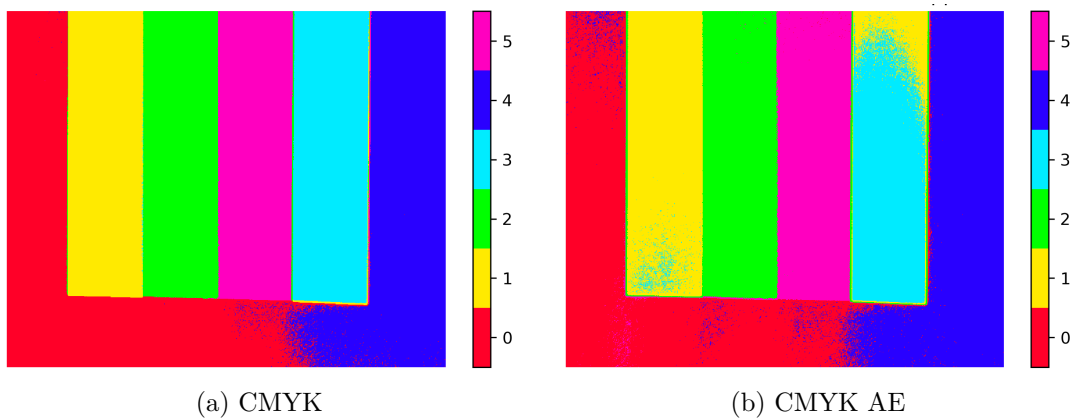


Figure 1.14: Comparison of  $k$ -means segmentation before and after encoding.

The colour labelling of each cluster (or class) is assigned randomly for each run of the  $k$ -means algorithm. For illustration purposes, the cluster colours were united for

these two evaluations. In general, the same assigned class (colour) on different input data does not mean that these classes share the same properties. The class order is also random and does not indicate any underlying patterns in the data. The  $k$ -means algorithm performed quite well on the entire MS image, aside from the necessity to increase the number of classes to counter the uneven illumination of the background. For the clustering of the AE compressed representation, the results are close to the full MS image input.

Each cluster can now be examined by its spectral property. The first approach is to calculate the mean reflectance spectrum of the cluster (Fig. 1.15).

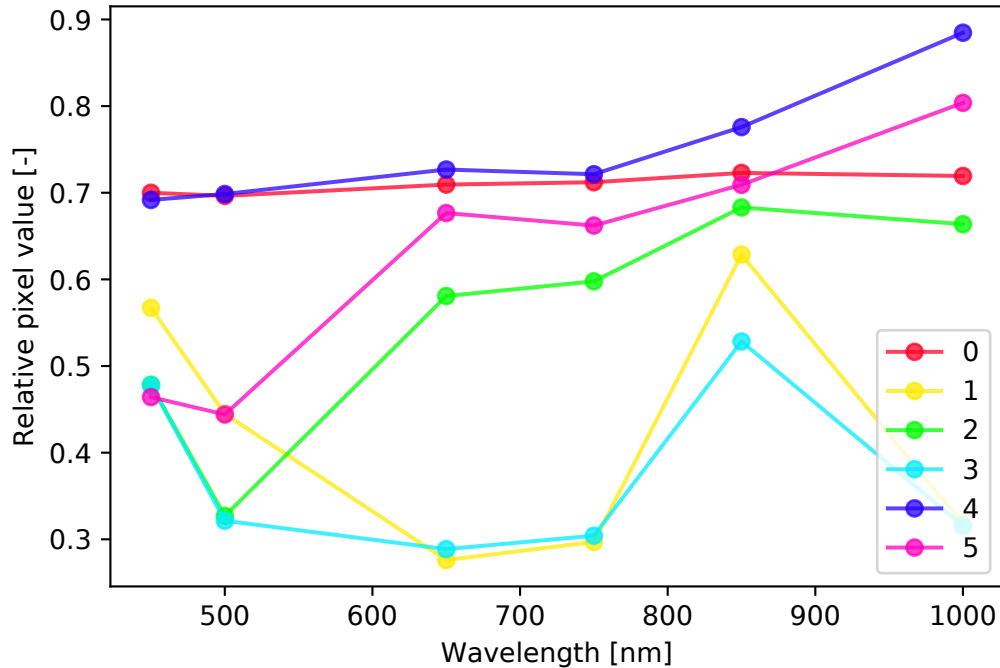


Figure 1.15:  $k$ -means average cluster spectra for CMYK data.

The colour labelling of the mean spectra matches the clusters in Fig. 1.14 (a). The spectral response of the cluster 0 and 4 are similar, confirming the fact that they form two subclasses of one common class – white background paper. By calculating only the mean value of each spectral band, the variance inside the band values is neglected. It is useful to plot each spectrum using a box plot to visualise the uncertainty inside the data (Fig. 1.16).

The box plot of each band is represented by its median (blue line), mean (red dashed line), lower and upper quartile ( $q_1$ ,  $q_3$  – size of the box) and whiskers ranging from  $q_1 - 1,5(q_3 - q_1)$  to  $q_3 + 1,5(q_3 - q_1)$  [16]. Both classes covering the background (0 and 4) exhibit a high variance in the 1000 nm band. The uneven illumination introduced additional noise and increased the uncertainty within this band.



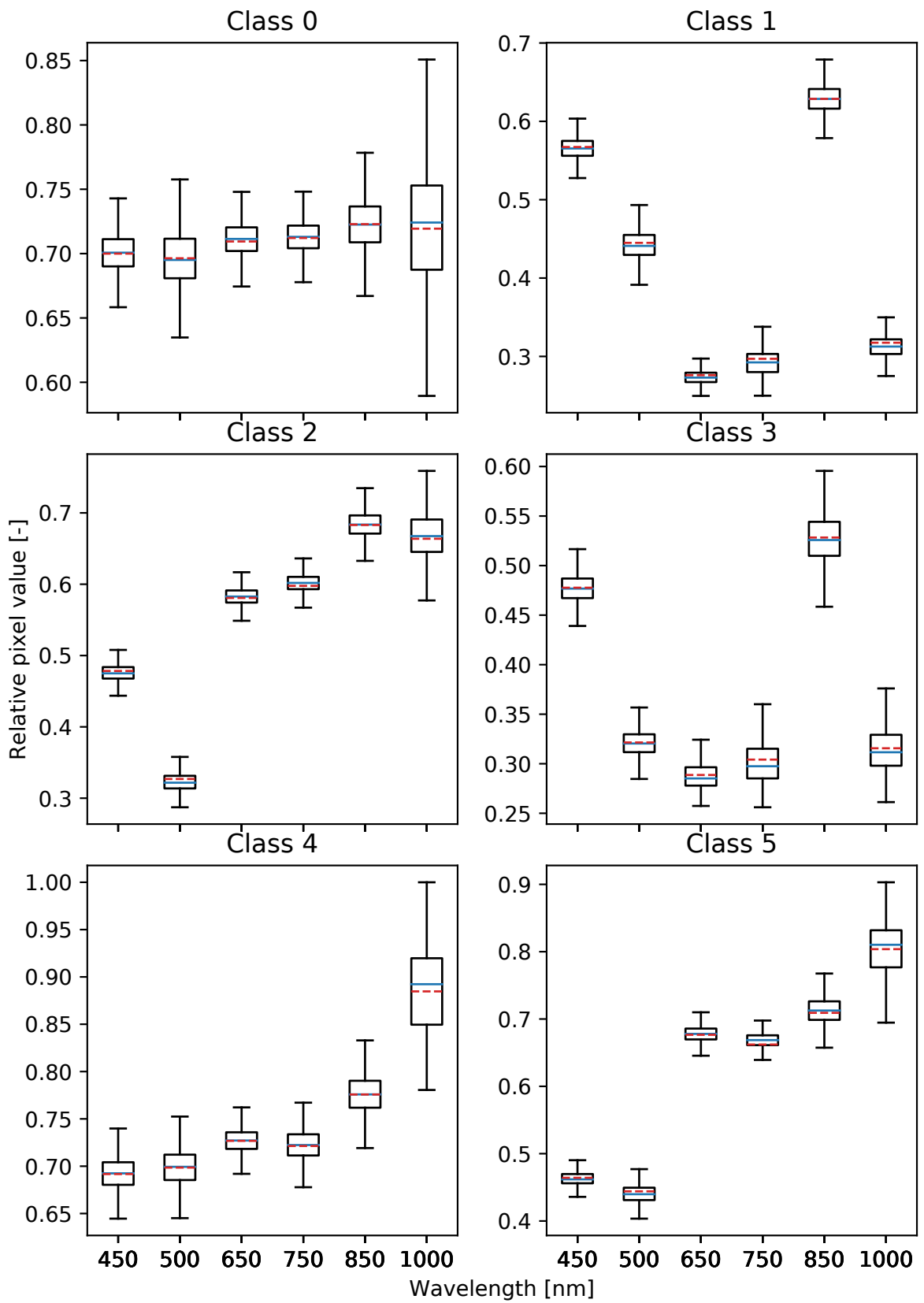


Figure 1.16:  $k$ -means box plot of individual classes for CMYK data.

## 1.5 Theoretical Part Conclusion

Spectral imaging provides a convenient way of measuring spectral response of objects while maintaining high spatial resolution. A multispectral image can be captured using a CCD camera with narrow-band spectral filters. The number of measured bands (spectral resolution) depends on the choice of bandpass filters, defined by their central wavelength and transmission bandwidth. With the aid of image calibration and normalisation techniques, final MS image is formed from individual images taken the CCD camera setup.

The visualisation of the captured data can be done by observing individual bands or band arithmetic by calculating specific remote sensing indices. Each index is constructed to highlight certain feature inside the image. This thesis focuses on plant vitality, so NDVI and CI-G indices are selected. More advanced analysis procedures take advantage of the entire MS image. Using  $k$ -means algorithm, the difference in the spectral response of the materials within the MS image can be identified. Combined with the ML method of data compression using an autoencoder, the MS image can be segmented into clusters of similar spectral response. These clusters can be further analysed by a specialist and their response can be matched with a reference spectral library

# Chapter 2

## Experimental Part

### 2.1 Overview

In the experimental part of this thesis, we focused on the MS image acquisition for the selected scene – park behind the Faculty of Science. The scene contained plants, namely grass and trees. We performed the measurement at the laboratory at the Department of Optics, with the camera pointing outside the window.

Firstly, we captured the MS images using a CCD camera paired with spectral filters mounted inside a custom 3D printed filter carousel. We acquired multiple exposures of each image for noise reduction. Then, we calibrated and normalised the MS images using a reference surface. We repeated the measurement for four different dates to explore the change in the spectral properties of the detected material.

Secondly, we utilised the MS analysis methods described in the theoretical part to extract information about grass vitality inside the scene. At first, we calculated the index values – NDVI and CI-G. Then we used  $k$ -means algorithm to analyse the entire MS image. We trained an autoencoder for each MS image to compress the spectral cube to two dimensional latent space. We again ran the  $k$ -means algorithm for the latent space variables and compare the outputs. The code used for processing including its outputs is presented in the text and appendices.

Finally, we performed the time series analysis of the four measurements. The index values, as well as spectral properties, are compared throughout the year. To conclude, we evaluated the capabilities and potential of the presented methods.

### 2.2 Multispectral Image Acquisition

The complete list of used tools for the MS image acquisition is as follows:

- Interference filter set (Carl Zeiss Jena)
- Filter carousel (3D printed)
- Lens MC Flektogon  $f = 20$  mm with  $2\times$  extender
- G2-4000 scientific CCD camera (Moravian Instruments)
- Computer with *Scientific Image Processing System* (SIPS) software
- Tripod, power supply and USB cable
- White sheet of paper (used as normalisation reference)

Figure 2.1 shows the experimental camera setup scheme for our measurement. For a photo of the completed setup in the laboratory, see Appendix 2.

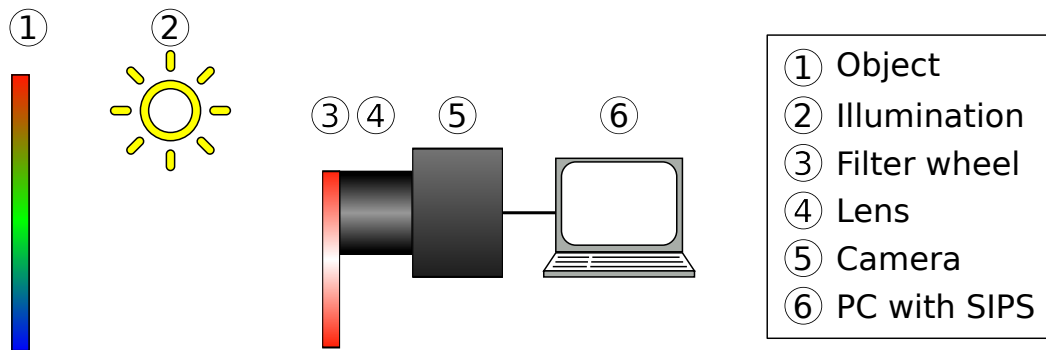


Figure 2.1: Experimental setup scheme.

To assess the grass vitality, we needed a reference spectral response of grass in different states. Using USGS spectral library [17], we plotted the reflectance spectra of fresh and dry grass (Fig. 2.2).

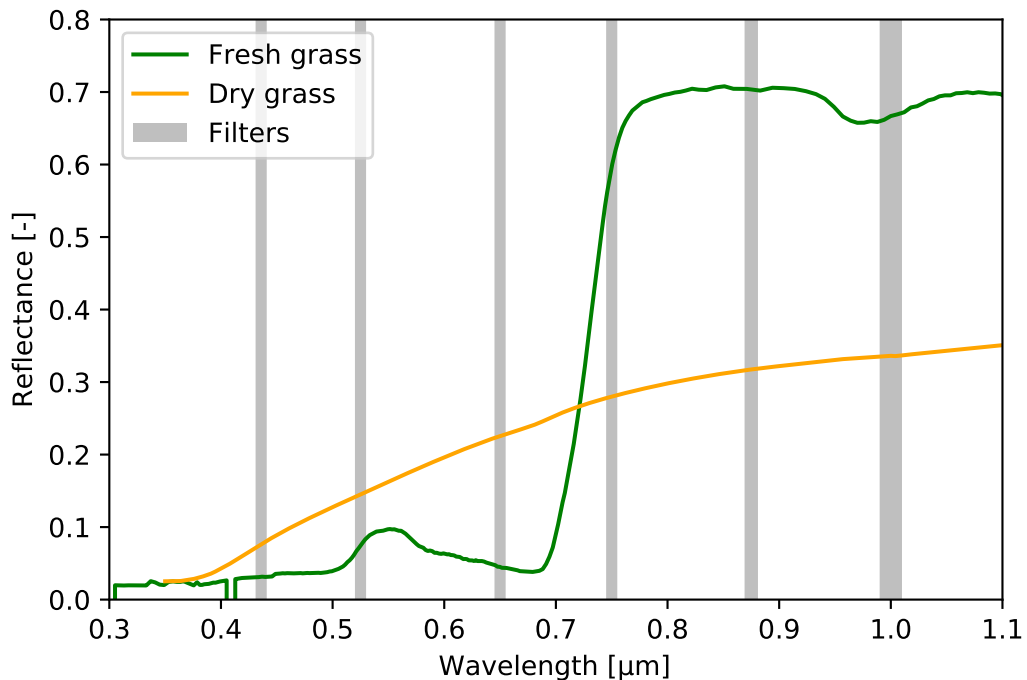


Figure 2.2: Fresh and dry grass reflectance spectra with used filters [18, 19].

The grey bars inside Figure 2.2 represent the spectral coverage of the filters selected for our MS image acquisition. The parameters describing the filters are following:  $\lambda_c$  central wavelength of filter,  $\Delta\lambda_c$  uncertainty in the position of central wavelength and FWHM full width at half maximum of the transmittance. The manufacturer of the used filters – Carl Zeiss Jena – identifies several types of filters. We use the Interference Filters (IF) and Special Interference Filters (SIF). The individual parameters of used filters are listed in Table 2.1.

Table 2.1: List of used spectral filters and their properties [15].

Filter name	$\lambda_c$ [nm]	$\Delta\lambda_c$ [nm]	FWHM [nm]	Label
SIF 436Hg	436	2	$10 \pm 2$	B
IF 525	525	5	$9 \pm 2$	G
IF 650	650	7	$9 \pm 2$	R
IF 750	750	8	$10 \pm 2$	-
IF 875	875	9	$13 \pm 3$	NIR
IF 1000	1000	10	22	-

The camera we used is equipped with an internal filter carousel but the diameter of the internal filter positions did not match the dimensions of filters wanted to use. Therefore we designed and 3D printed filter carousel, that could be mounted onto the lens. The design was modelled using *Autodesk Fusion 360*. The model of the filter carousel is presented in Figure 2.3.

The final form of the filter carousel was slightly modified in contrast with the design, mainly the filter position locking mechanism. Instead of using spring-loaded stopping bolts, the position of the carousel is fixed using small neodymium magnets. I would like to thank RNDr. Jaroslav Wagner Ph.D. for this improvement and help with the overall realisation of the 3D print.

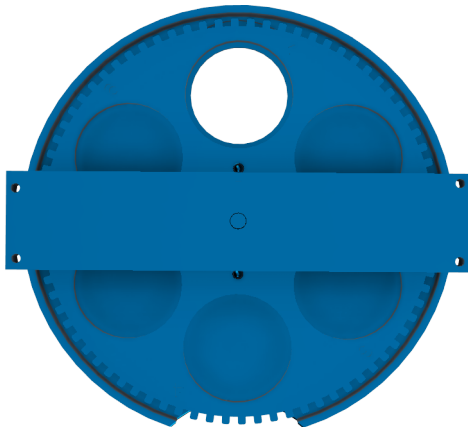


Figure 2.3: CAD model of the custom filter carousel.

The G2-4000 camera is chosen for its low SNR, cooling capabilities, high resolution ( $2056 \times 2062$ ), large pixel size ( $7,4 \times 7,4 \mu\text{m}$ ) and 16-bit ADC resolution. All of these factors are important for long exposure detection of weak signals.

The lens was mounted onto the camera using  $2\times$  extender adaptor, effectively changing the focal length to  $f' = 40$  mm. The lens aperture was set to  $f/11$  to achieve optimal sharpness of the image while not limiting the amount of incoming light.

When choosing the exposure time, the goal was to prevent under or overexposure (saturation of the sensor). Different filters required various exposure times due to the dependence of the CCD sensitivity on the incoming wavelength (Fig. 1.2), as well as the spectral composition of the illumination of the scene (Fig. 1.3). All these effects are compensated using image normalisation (Eq. 1.2).

The step by step procedure for image acquisition is discussed in Section 1.2.1. We selected a scene that contains a part of the park behind the Faculty of Science – lawn with grass and trees (Fig. 2.4).



Figure 2.4: Overview of the detected scene – park behind the Faculty of Science (composed from the 19/06/04 measurement, where R = IF 650, G = IF 525 and B = SIF 436Hg).

Each image exposure was repeated 10 times for averaging. Each measurement, therefore, requires taking 60 light exposures, 10 flat field images and up to 60 dark frames (depending on the exposure times of the light images). To assess the grass vitality throughout the year, we decided to make four measurements. This gives us a total of 240 raw images, plus required calibration images. The measurements took place on dates recorded in Table 2.2. The dates are listed together with the weather conditions for each measurement. Each weather condition parameter is a mean value for the entire day, collected from a weather station located in street U kovárny, Olomouc.

Table 2.2: Measurement dates and weather conditions [20].

Date [YY/MM/DD]	Temperature [°C]	Humidity [%]	Pressure [hPa]
19/02/28	6.3	74	1011.1
19/04/04	14.6	53	1002.1
19/06/04	22.0	64	1010.9
19/10/30	3.5	79	1027.7

The image calibration and normalisation procedures are summarised in Figure 2.5. Averaging, flat field and dark image corrections were done on each image using SIPS software. The images were normalised using Python script `0-preprocessing.py`. The main part of this code is presented in Appendix 3. The entire code used in this thesis is available in public repository [21].

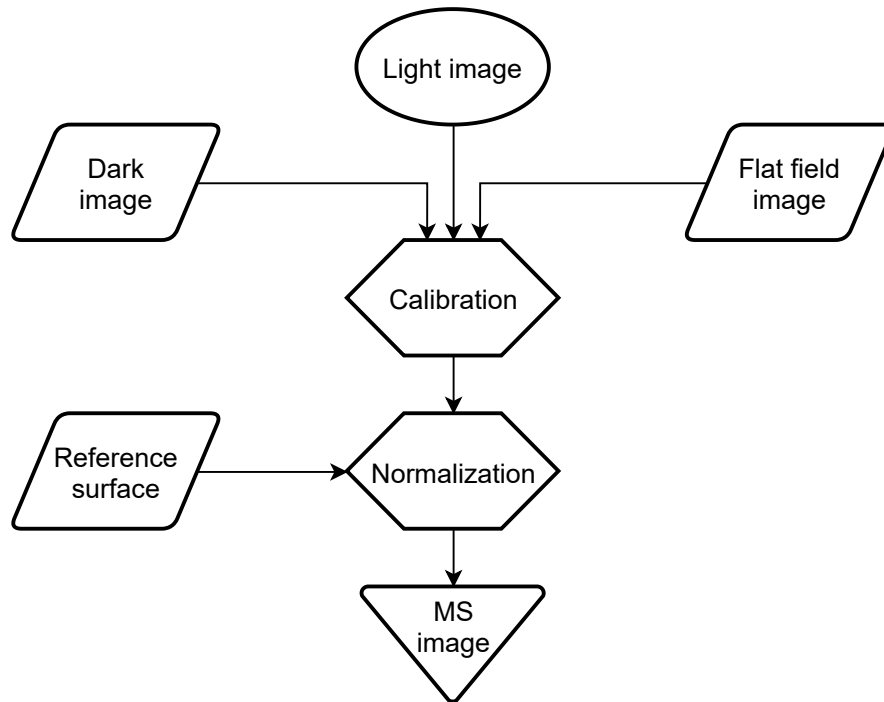


Figure 2.5: MS image preprocessing flowchart.

Figure 2.6 presents the scene detected with IF 525 filter, after calibration and normalisation. All four MS images together form a dataset that is ready for further analysis.

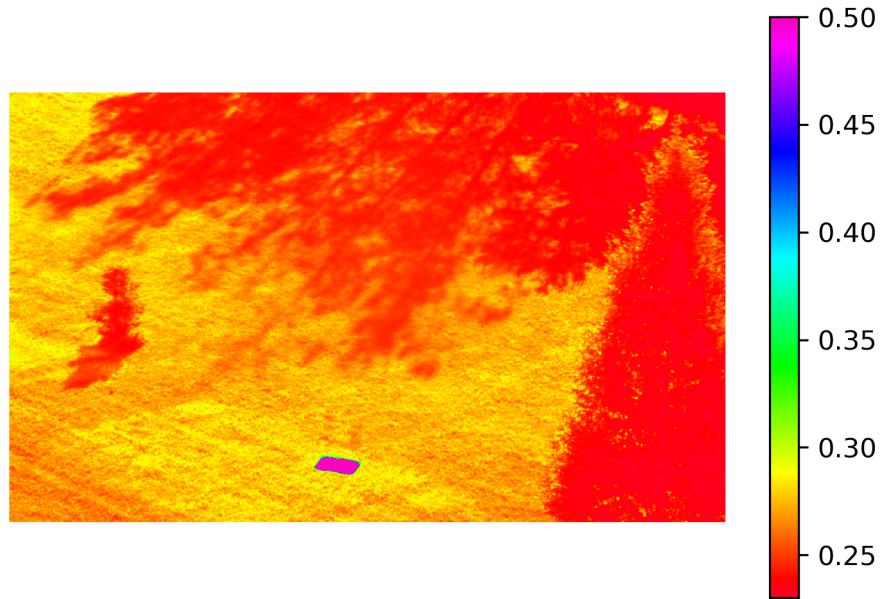


Figure 2.6: Detected scene via the IF 525 filter, visualised using false colour (19/06/04 measurement, cropped).

## 2.3 Multispectral Image Analysis

Now we aim to analyse the MS image dataset we created. The individual methods used for the analysis are summarised in Figure 2.7.

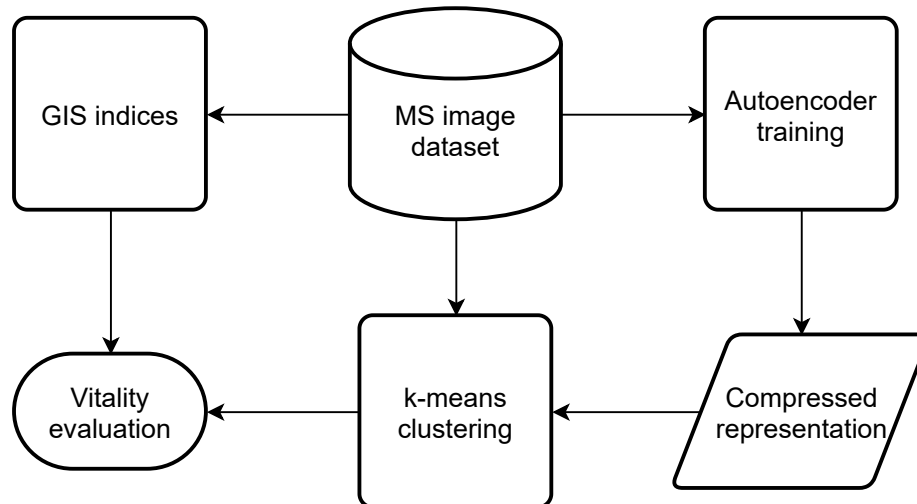


Figure 2.7: MS image analysis flowchart.

### 2.3.1 Remote sensing indices analysis

We start with the calculation of remote sensing indices, namely NDVI and CI-G. We evaluate the index value for each pixel using equations 1.3 and 1.4. The implementation of this procedure is located in file `indices.py`, see Appendix 4. Example of NDVI index calculated for measurement 19/06/04 is presented in Figure 2.8.



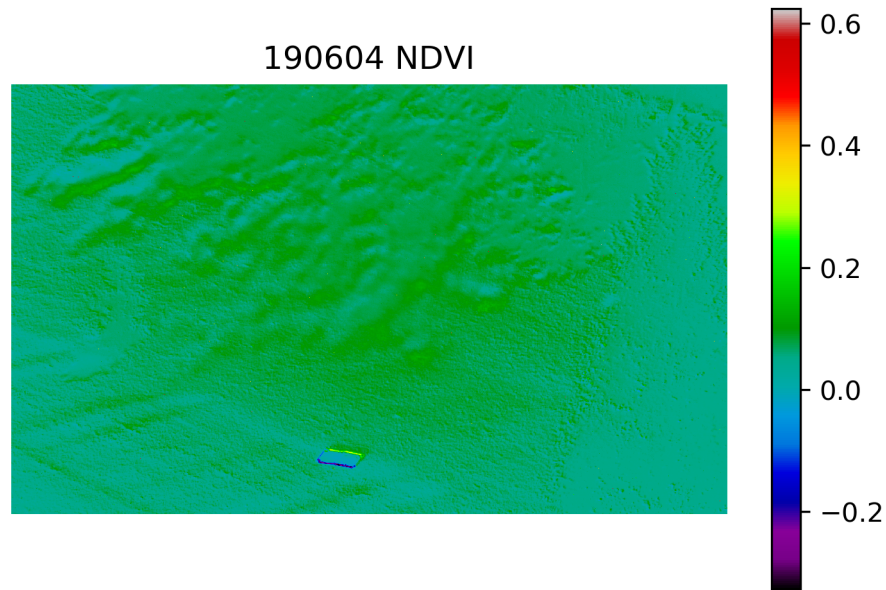


Figure 2.8: NDVI for 19/06/04 measurement.

The index visualisation consists of mostly positive values of the NDVI, signalling the presence of green plants. To evaluate the distribution of the index values, it is useful to plot a histogram of the index values (Fig. 2.9). The NDVI values range from 0.02 to 0.12 and do not change dramatically within the scene.

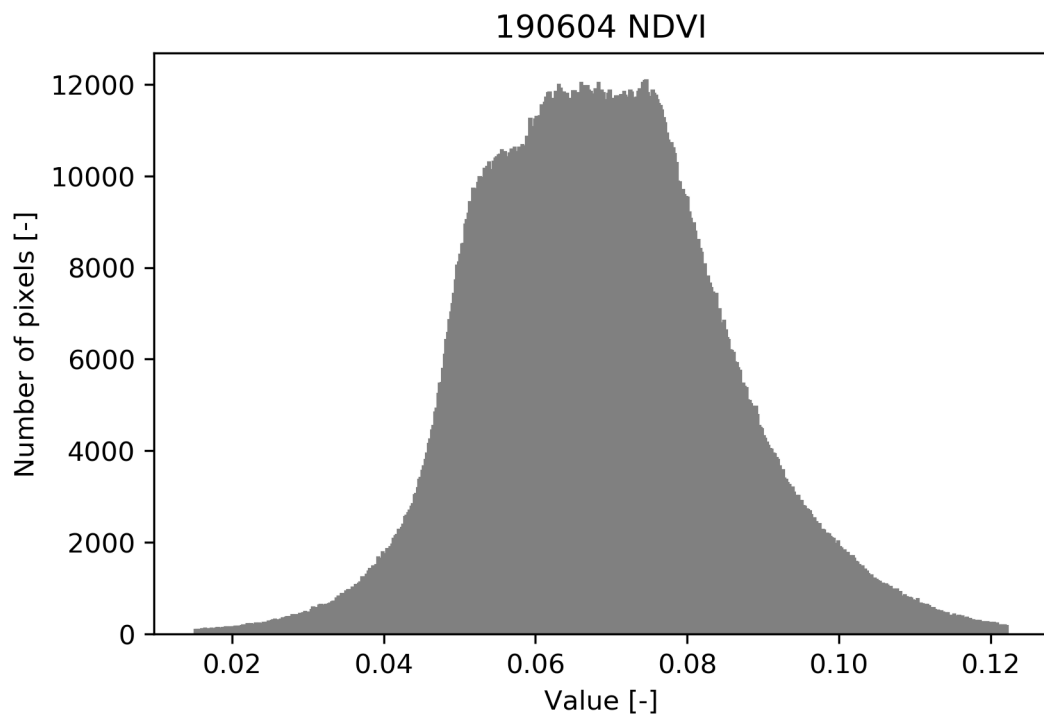


Figure 2.9: Histogram of NDVI for 19/06/04 measurement.

As we can see, the CI-G index is quite uniform in the image, with positive values signalling the presence of chlorophyll (Fig. 2.10).

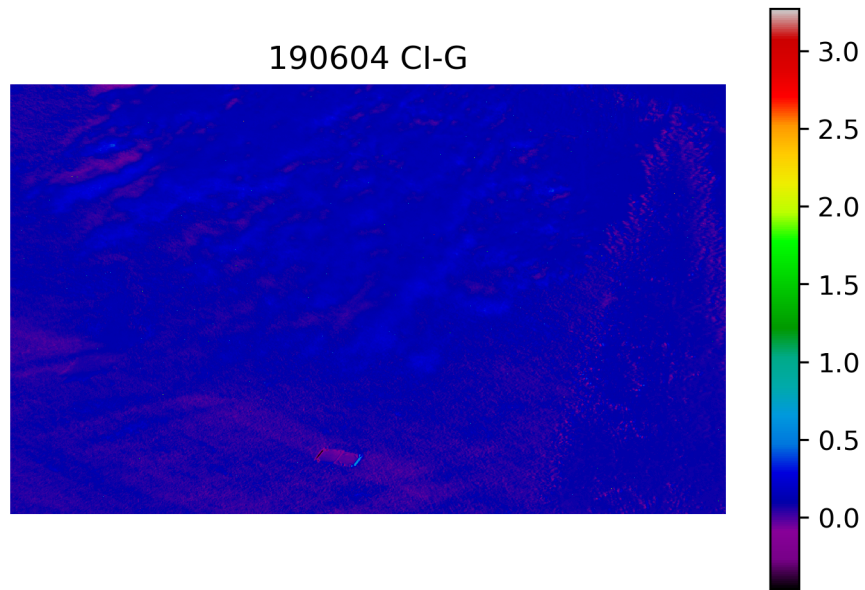


Figure 2.10: CI-G for 19/06/04 measurement.

The histogram shows that the variance of the CI-G is low as well (Fig. 2.11).

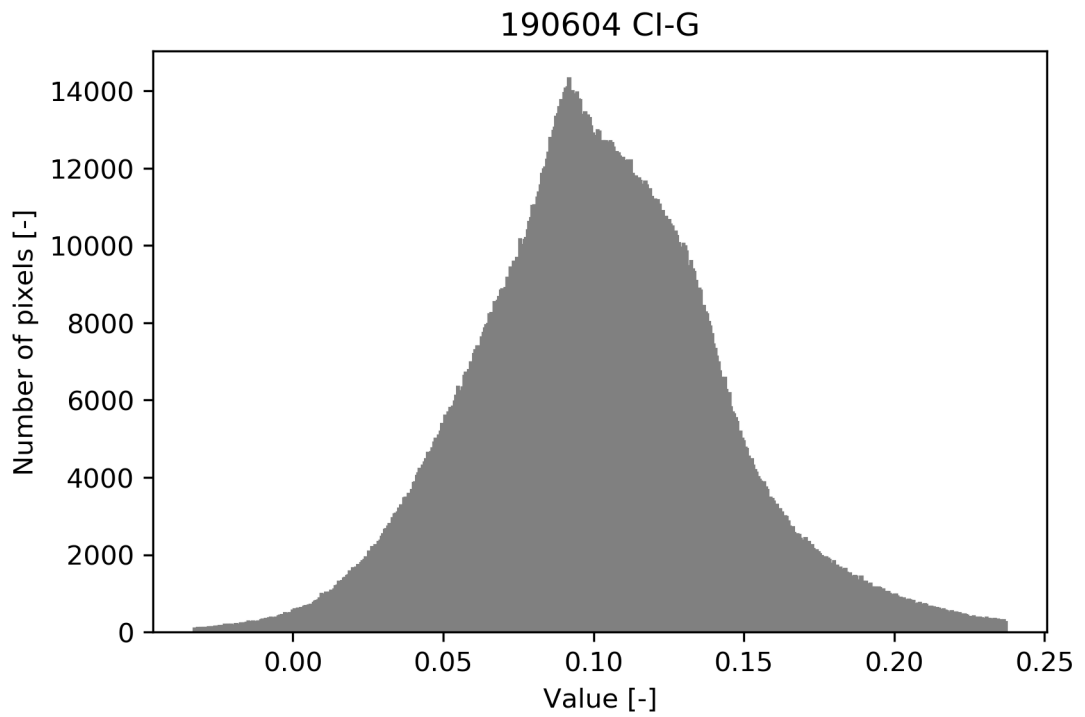


Figure 2.11: Histogram of CI-G for 19/06/04 measurement.

### 2.3.2 $k$ -means image segmentation

The  $k$ -means algorithm implementation utilises the *scikit-learn* machine learning Python module [22]. When we process the entire MS image using  $k$ -means algorithm, more detailed differences start to appear inside the scene. Firstly we evaluate the required number of clusters via the elbow method (Fig. 2.12).

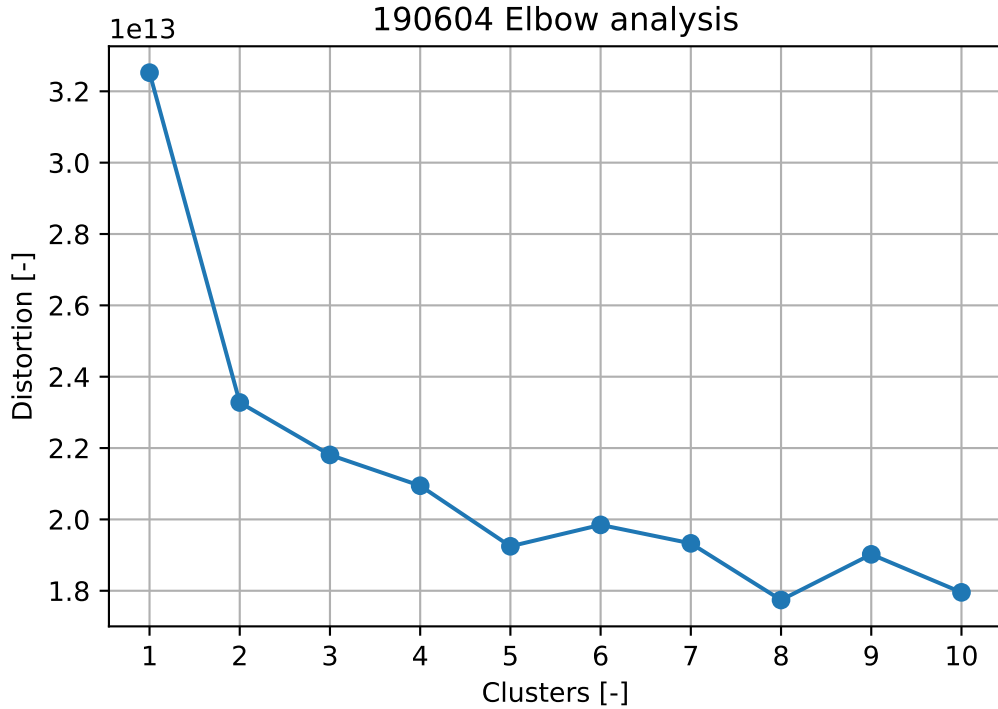


Figure 2.12:  $k$ -means elbow analysis of the 19/06/04 measurement.

The elbow analysis curve does not contain any significant elbow point, therefore we choose the value with the lowest inertia,  $k = 8$ . For our example, the number of clusters can be higher, because the identified classes can be, after further analysis, labelled as two subclasses of one common class. The elbow analysis for the remaining measurements is presented in Appendix 5. After the image segmentation, more nuanced differences inside the image become apparent (Fig. 2.13). For  $k$ -means clustering output for the rest of the measurements, see Appendix 6.

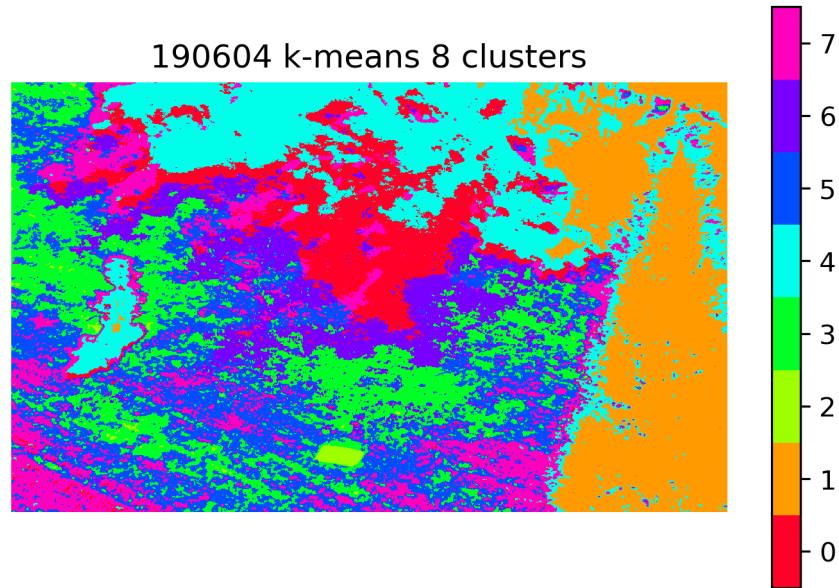


Figure 2.13:  $k$ -means identified clusters for the 19/06/04 measurement.

The  $k$ -means clustering separated the white normalisation surface (cluster 2). It also separated the tree located in the right part of the image (cluster 1). For each cluster, we identified the mean spectral response (Fig. 2.14), which highlight the differences between them.

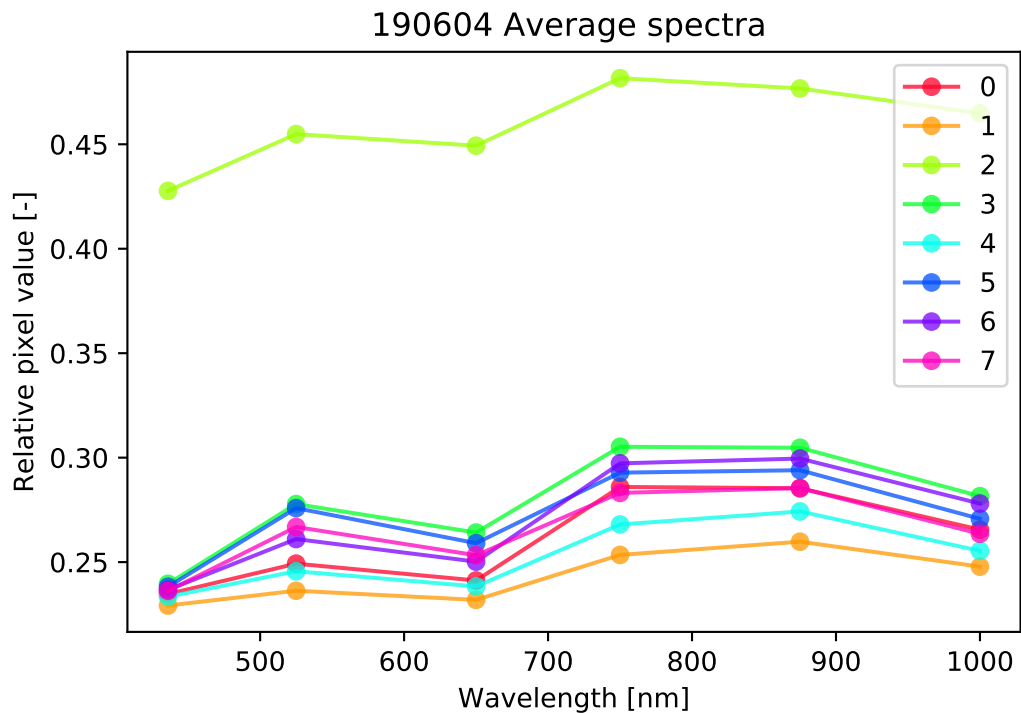


Figure 2.14:  $k$ -means mean spectra per class for the 19/06/04 measurement.

Class 2 represents the spectral response of the white normalisation surface. Class 1 captures the spectral reflectance of the tree located on the right side of the scene. For better visualisation of the reflectance spectra, we plotted the spectral response for each class separately using a box plot (Fig. 2.15). The box plot reveals the variance

inside each detected band. The solid blue line inside the box shows the median value and red dashed indicates the mean value of the spectral band.

The variance inside class 2 is high due to the high reflectance of the normalisation surface, combined with clipping values of some detected pixels (some whiskers reach to the maximum relative pixel value). The remaining classes exhibit spectral response similar to fresh grass depicted in Figure 2.2. This indicates the presence of photosynthetic plants. The specific condition of the detected plants would require field tests of chlorophyll concentration to match it with the spectral response or further analysis of the spectra by a specialist.

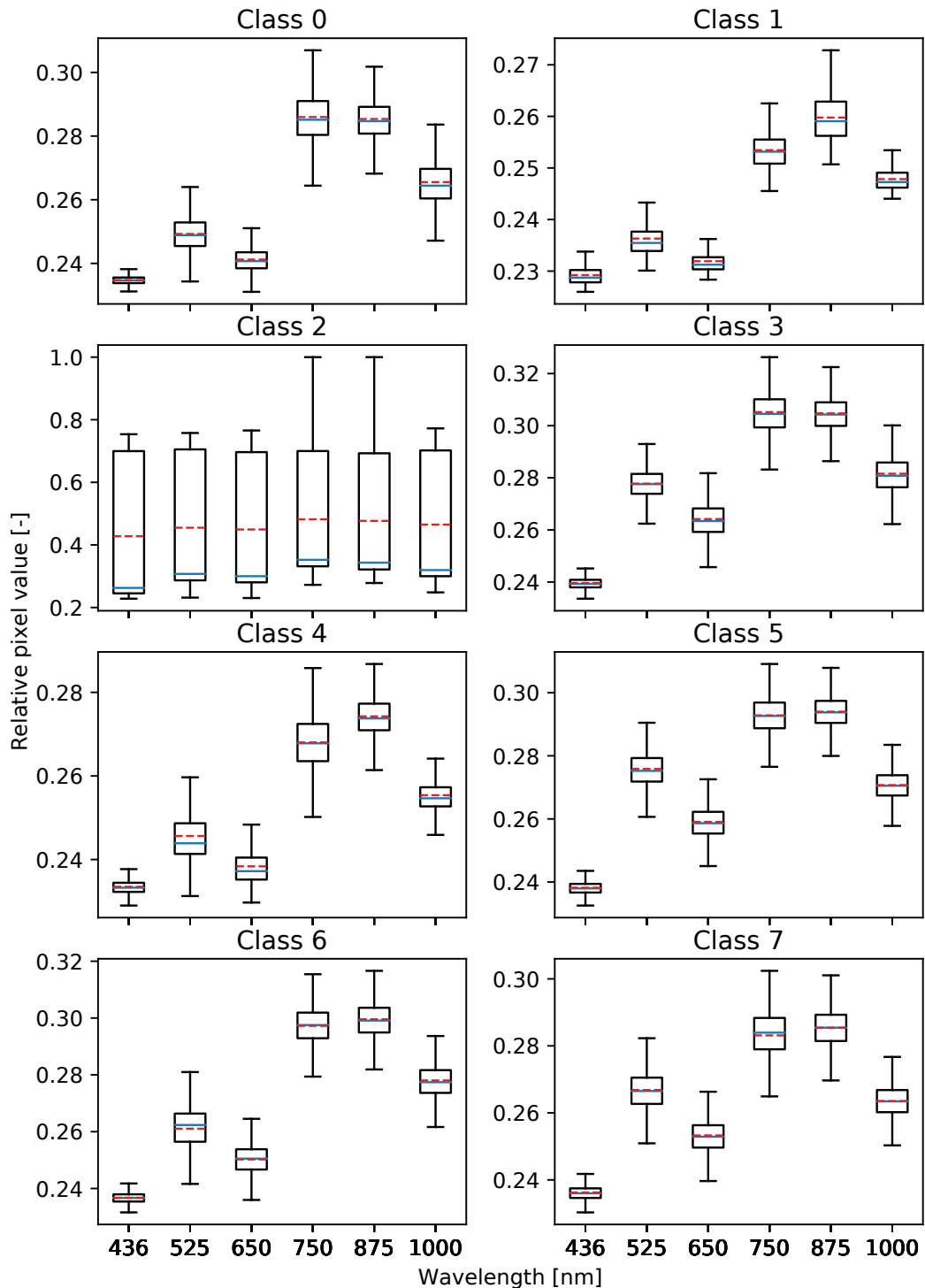


Figure 2.15: Spectrum box plot per class of the 19/06/04 measurement.

### 2.3.3 Autoencoder for dimensionality reduction

We take advantage of the autoencoder to obtain a two-dimensional compressed representation of the MS image. Each MS picture was converted to a set of vectors, representing the spectral response of each pixel inside the image. This set was then used as a training input for autoencoder neural network. The network design is identical to the one discussed in section 1.4.

The main part of the autoencoder script is presented in Appendix 7. We used the Keras API for the neural network building and training [13]. The input data was split in 80:20 ratio, 80% for training and 20% for testing. We set the batch size to 2048 and we train for a total number of 10 epochs. The `1b-autoencoder.py` script imports the MS data, trains the autoencoder, outputs the encoded form of the MS image and saves the trained model.

The resulting latent space after  $k$ -means classification is presented in Figure 2.16.

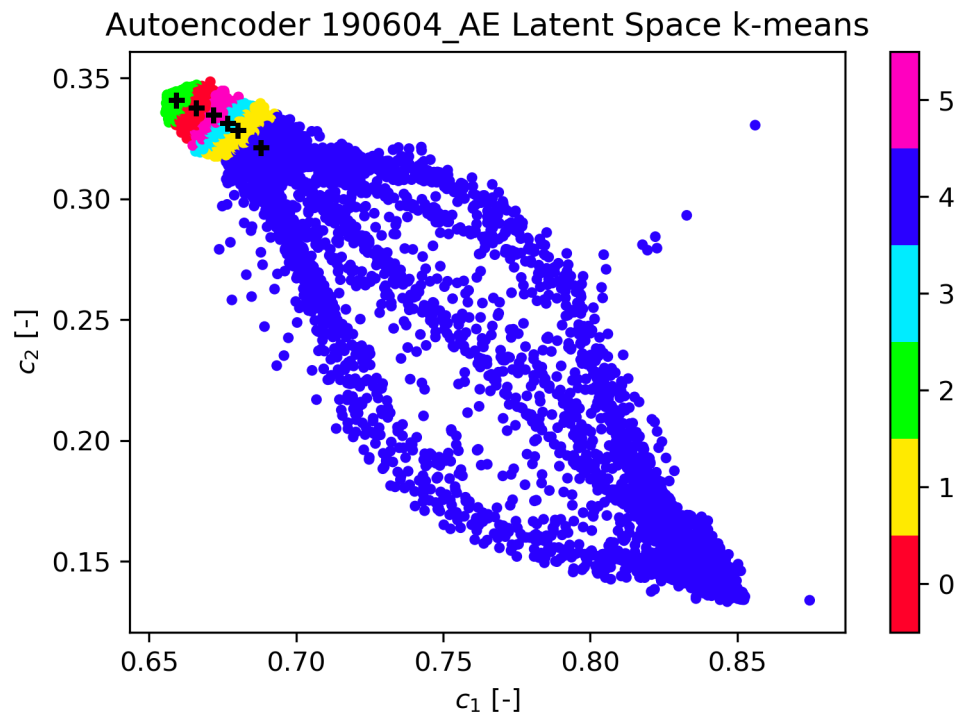


Figure 2.16: Latent space of the 19/06/04 measurement with highlighted clusters for  $k = 6$ .

The latent space variables  $c_1, c_2$  do not represent any measured value, therefore they are dimensionless. After the  $k$ -means clustering, the visualisation of the identified clusters inside the detected scene has similar properties as the processing of the entire MS image (Fig. 2.17).

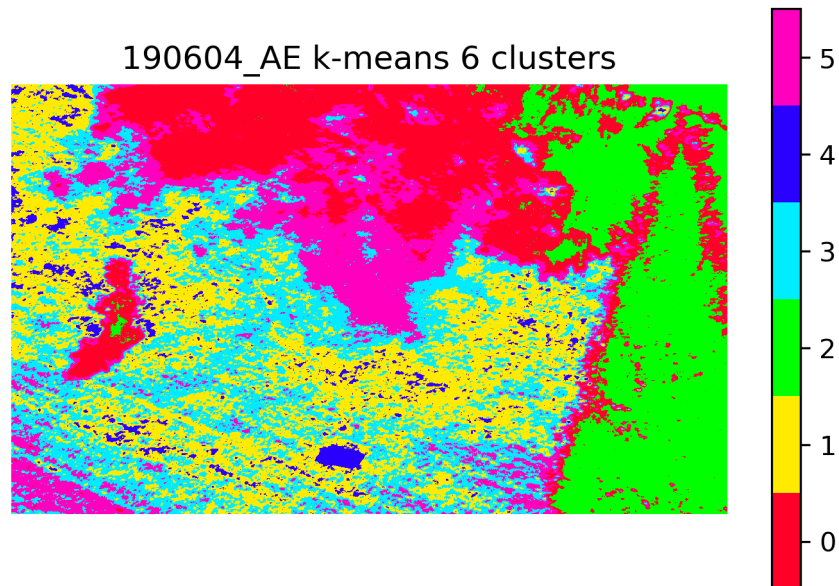


Figure 2.17: Identified clusters for the 19/06/04 measurement after compression.

The compressed representation of the MS image still contains enough information to separate the normalisation paper (cluster 4) as well as the tree located on the right side of the frame (cluster 2). To compare the result better, we plotted a cropped part of the clustered image before and after the input data compression (Fig. 2.18).

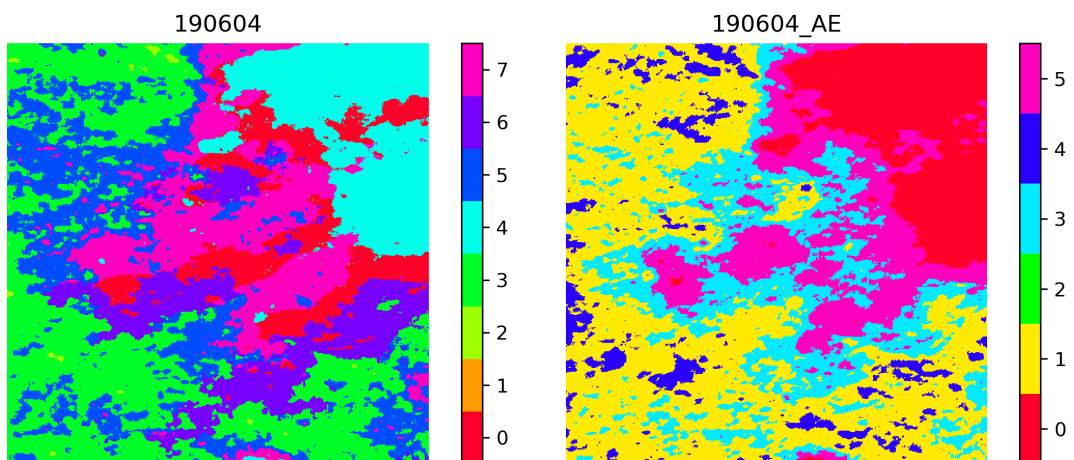


Figure 2.18: Comparison of the  $k$ -means clustering output on entire dataset and compressed representation (19/06/04 measurement, cropped).

The elbow analysis of the compressed representation resulted in setting the  $k = 6$ , resulting in lower separation of the input data. However, it is possible to set the  $k$  value equal for both clustering processes.

## 2.4 Time Series Comparison

The index values allow a direct numerical comparison between the measurements. The box plot of the NDVI values inside the entire image for each measurement is presented in Figure 2.19 a. The CI-G index exhibits similar changes (Fig. 2.19 b).

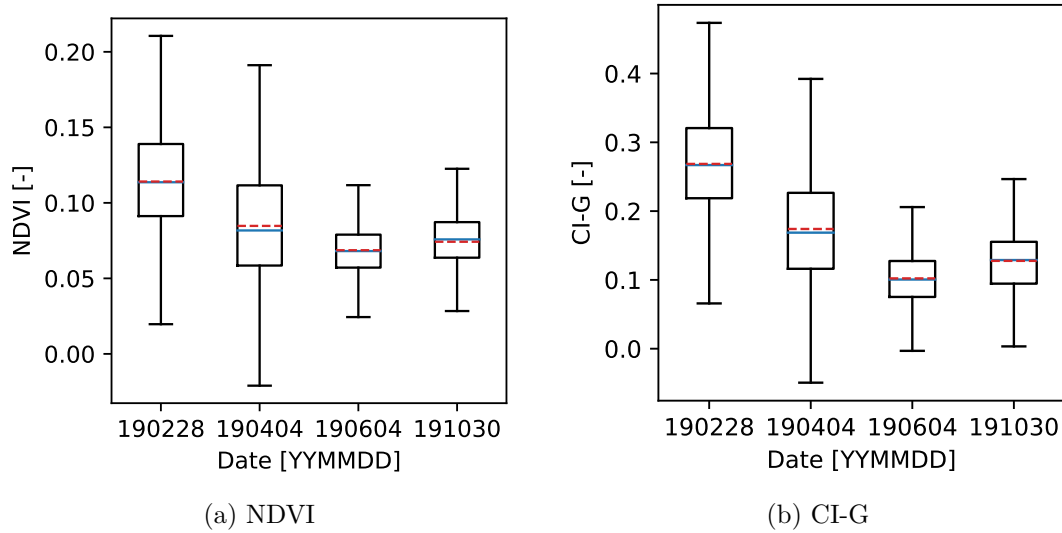


Figure 2.19: Box plot of index values for all measurements.

The indices values vary between the measurements, the mean values of both NDVI and CI-G index reach higher values during the first and second measurement and drops for the following two. If we look at the cropped parts of the index image, the change in index values is apparent (Fig. 2.20).

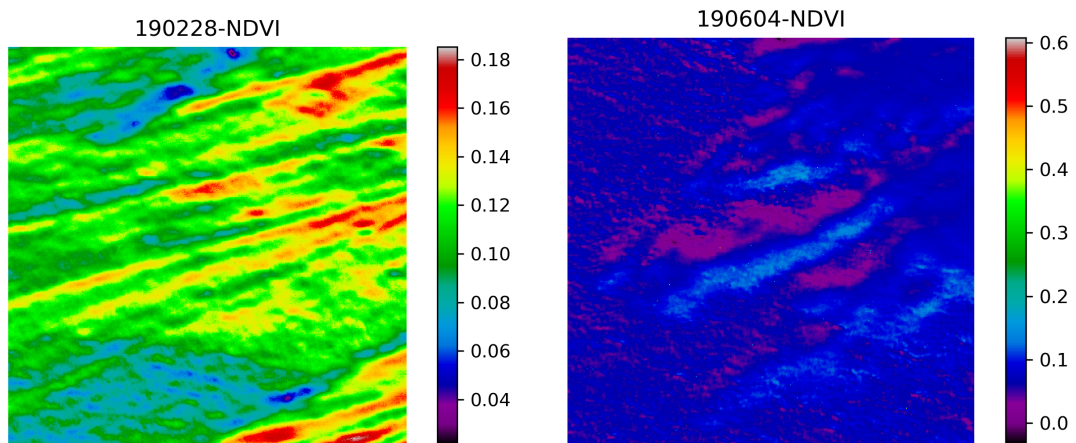


Figure 2.20: Comparison of the NDVI values for two measurements, cropped.

The reason behind this may be a better condition for the grass during more humid months of February and April in comparison with dry and hotter June or October.

If we look at the clustered images, we can compare the spectral response of selected classes. Figure 2.21 shows a cropped part of two images after  $k$ -means clustering.



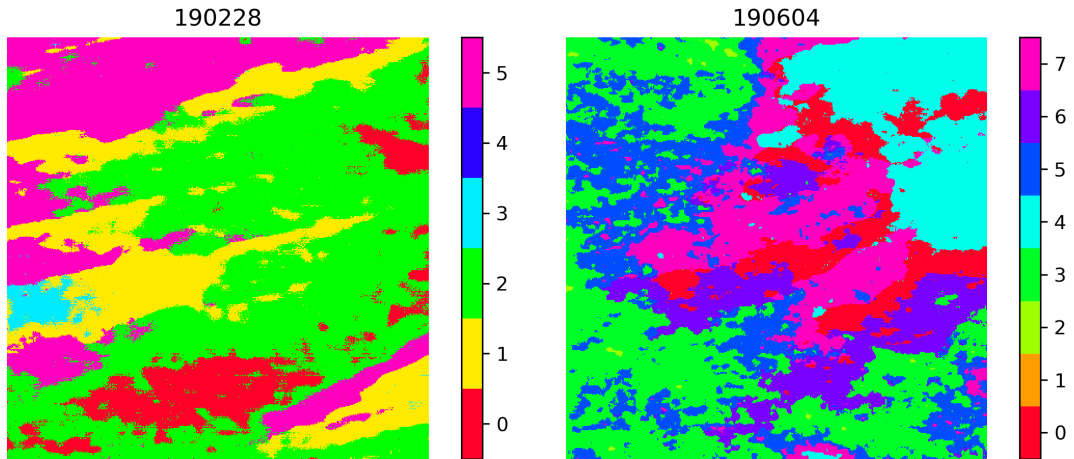


Figure 2.21: Comparison of the labelled images for two measurements, cropped.

For spectral comparison, we selected class 1 from measurement 19/02/28 and class 6 from measurement 19/06/04 (Fig. 2.22).

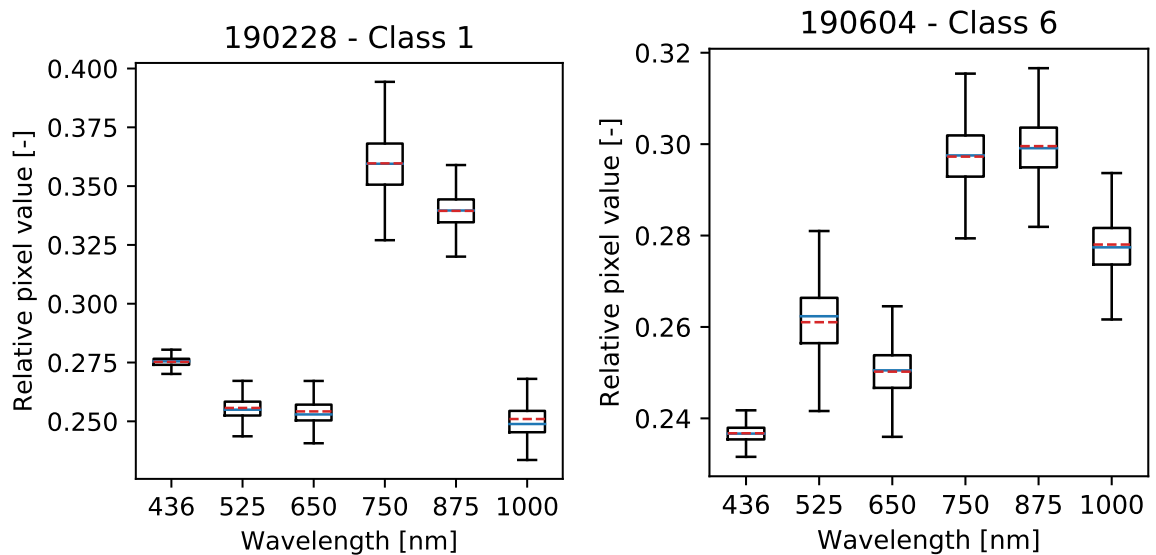


Figure 2.22: Comparison of selected classes spectrum box plot.

The spectral box plots of each of the selected classes share features similar to a spectral response of grass presented in Figure 2.2. Values for bands below 650 nm exhibit lower reflectance than the bands from 750 nm up. The difference between measured values in these two regions is much larger for the 19/02/28 measurement than for the 19/06/04. This indicates that the grass was in better condition during the first measurement.

## 2.5 Experimental Part Conclusion

Using a CCD camera with active cooling we obtained raw images, which were then calibrated and normalised to obtain the final MS images (Fig. 2.23 a). We repeated the MS acquisition in four different days throughout the year. We aimed to estimate the plant vitality in the chosen scene – park behind the Faculty of Science.

Firstly we analysed the MS image using NDVI (Fig. 2.23 b) and CI-G indices, both of them yielding positive values indicating the presence of chlorophyll with a low variation inside the scene. Secondly, we used the  $k$ -means clustering algorithm to analyse the entire MS image (Fig. 2.23 c). The output classes can be analysed by their spectral response (Fig. 2.23 d).

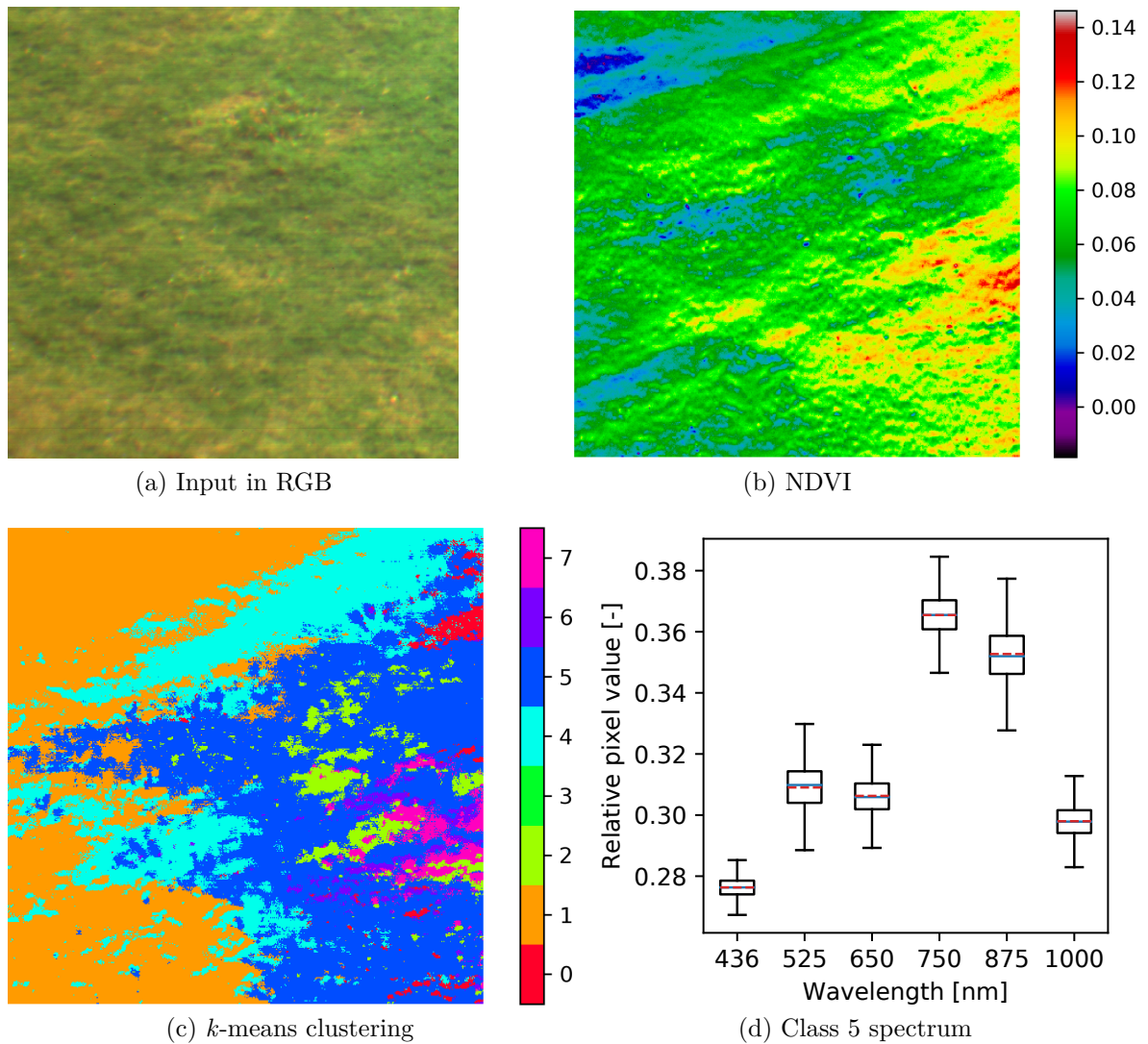


Figure 2.23: MS image analysis summary.

Using an autoencoder, we created a compressed representation of the data, which was then also processed by the  $k$ -means algorithm. The compressed data still retained most of the information contained inside the entire MS image (Fig. 2.18). This method may become particularly useful when dealing with MS images with a large number of bands or hyperspectral images. Lastly, we compared the results of the analysis for all measurements. The index values varied with time (Fig. 2.19), suggesting the impact

of different conditions on the plant vitality. Accurate conclusions regarding the grass vitality would require further analysis.

The output of proposed MS image analysis can be interpreted by an expert in the field connected to the objects inside the observed scene, in this case, plant biology or agriculture. The measured data can be combined with field tests of chlorophyll contents to quantify the vitality of the plant. The outputs can also be used for plant phenotyping, disease and species identification [23].

# Conclusion

The theoretical part of this thesis introduced an overview of the MS image acquisition technique and three selected approaches for the MS image analysis. Firstly, remote sensing indices were used to evaluate the specific properties of observed objects. Secondly, spectral clustering was accomplished utilising the  $k$ -means algorithm to identify areas with similar spectral properties. Lastly, by applying an autoencoder neural network, data was converted to compressed representation, which was then also clustered using  $k$ -means algorithm.

The experimental part described steps taken to acquire the MS images of the selected scene, which contained grass and trees. The measurement took place over four different days during the year. The goal of the MS data analysis was to estimate the plant vitality. Measured data were analysed using techniques discussed in the theoretical part. Results indicate that different conditions throughout the year affect the vitality of the observed plants. A more accurate conclusion would require an expert evaluation.

The advantage of the proposed MS image processing is independence on the number and type of selected filters (spectral bands). Additionally, there is no need for prior information about observed materials. The disadvantage is that results could be influenced by the random initialisation of the  $k$ -means clustering algorithm. Furthermore, improper design or training of the autoencoder can result in a distorted compressed representation of the input MS image (the encoder outputs values for one compressed variable and the other is always coded to zero, effectively reducing the representation to one dimension).

The acquisition, as well as the analysis, can be automated and input data can come from any MS or HS imaging sensor. The use case is certainly not limited to biology or agriculture. As the MS imaging technology is becoming more available, many new applications are being tested and implemented. Methods and techniques presented in this thesis can be used as a foundation for further research.

# List of Abbreviations and Symbols

ADC	.....	Analog-to-digital converter
CCD	.....	Charge-coupled device
CI-G	.....	Chlorophyll index green
CMYK	.....	Cyan, magenta, yellow, key (black)
FITS	.....	Flexible Image Transport System
FWHM	.....	Full width at half maximum
HS	.....	Hyper-spectral
IF	.....	Interference filter
ML	.....	Machine learning
MS	.....	Multi-spectral
MSE	.....	Mean squared error
NDVI	.....	Normalised difference vegetation index
NIR	.....	Near infrared
NN	.....	Neural network
Pixel	.....	Picture element
RGB	.....	Red, green, blue
SGD	.....	Stochastic gradient descend
SIF	.....	Super interference filter
SIPS	.....	Scientific Image Processing System
SNR	.....	Signal to noise ratio
$\mathbf{c}$	.....	Compressed representation vector
$C$	.....	Loss function
$d$	.....	Distance
$D$	.....	Distortion
$k$	.....	Number of clusters, input parameter for the $k$ -means algorithm
$\mathbf{x}$	.....	Data-point vector
$\lambda_c$	.....	Central wavelength of filter
$\Delta\lambda_c$	.....	Uncertainty in the position of the central wavelength
$\boldsymbol{\mu}$	.....	Cluster centre vector

# Bibliography

- [1] R. Paschotta, *Multispectral Imaging*. RP Photonics Encyclopedia (online, 2012). Available at [https://www.rp-photonics.com/multispectral\\_imaging.html](https://www.rp-photonics.com/multispectral_imaging.html)
- [2] O. Vaculík, *Využití senzoru v okrajových pásmech citlivosti*. Bachelor Thesis, (Olomouc, 2018). Available at <https://theses.cz/id/rkg6x1/BP-Vaculik-OPT-2018.pdf>
- [3] *KAI-04022: 2048 (H) x 2048 (V) Interline CCD Image Sensor*, ON SEMI-CONDUCTOR (online, 2015). Available at: <http://www.onsemi.com/pub/Collateral/KAI-04022-D.PDF>
- [4] *Introduction to CCD Imaging*. Moravian Instruments: CCD & CMOS Cameras For Astronomy (online, 2013). Available at <https://www.gxccd.com/art?id=303&cat=52&lang=409>
- [5] *ASTM G-173: Reference Air Mass 1.5 Spectra*. The National Renewable Energy Laboratory, U.S. Department of Energy (online, 2003). Available at <https://www.nrel.gov/grid/solar-resource/spectra-am1.5.html>
- [6] *Satellite image of Olomouc*, Sentinel-2 L1C, Sentinel Hub Playground (online, 2020). Available at [https://apps.sentinel-hub.com/sentinel-playground/?source=S2&lat=49.59162979415221&lng=17.262182235717773&zoom=13&preset=2\\_COLOR\\_INFRARED\\_VEGETATION\\_&layers=B01,B02,B03&maxcc=0&gain=1.0&gamma=1.0&time=2019-10-01%7C2020-04-02&atmFilter=&showDates=true](https://apps.sentinel-hub.com/sentinel-playground/?source=S2&lat=49.59162979415221&lng=17.262182235717773&zoom=13&preset=2_COLOR_INFRARED_VEGETATION_&layers=B01,B02,B03&maxcc=0&gain=1.0&gamma=1.0&time=2019-10-01%7C2020-04-02&atmFilter=&showDates=true)
- [7] F.J. Kriegler, W.A. Malila, R.F. Nalepka, and W. Richardson. *Preprocessing transformations and their effects on multispectral recognition*. Proceedings of the Sixth International Symposium on Remote Sensing of Environment, p. 97-131 (1969).
- [8] E. R. Hunt, C. S. T. Daughtry, J. U. H. Eitel, and D. S. Long, *Remote Sensing Leaf Chlorophyll Content Using a Visible Band Index*. Agronomy Journal, vol. 103, no. 4, pp. 1090-1099, (2011).
- [9] J. M. Dixon, M. Taniguchi and J. S. Lindsey, *PhotochemCAD 2. A Refined Program with Accompanying Spectral Databases for Photochemical Calculations*, Photochem. Photobiol., 81, 212-213 (2005). Available at <https://omlc.org/spectra/PhotochemCAD/html/123.html>
- [10] N. Shukla, *Machine Learning With TensorFlow* (Manning, Shelter Island, 2018).
- [11] S. Marsland, *Machine Learning: An Algorithmic Perspective, 2nd ed.* (CRC Press, Boca Raton, Fla., 2009).

- [12] D. J. C. MacKay, *Information Theory, Inference, And Learning Algorithms*, Reprinted with corrections (Cambridge University Press, Cambridge, 2004). Available at <http://www.inference.org.uk/itprnn/book.pdf>
- [13] Chollet, François *et al.*, *Keras: The Python Deep Learning library* (online, 2015). Available at <https://keras.io>
- [14] M. A. Nielsen, *Neural Networks and Deep Learning* (Determination Press, 2015). Available at <http://neuralnetworksanddeeplearning.com/>
- [15] *Metallinterferenzfilter*, Carl Zeiss JENA (online). Available at: <http://www.mikroskop-online.de/Mikroskop%20BDA%20Gruppe%2032/32-846b-1%20%20Metallinterferenzfilter.pdf>
- [16] Hunter, J., Dale, D., Firing, E. & Droettboom, M. *matplotlib.pyplot.boxplot*. Matplotlib (online, 2018). Available at: [https://matplotlib.org/3.1.1/api/\\_as\\_gen/matplotlib.pyplot.boxplot.html](https://matplotlib.org/3.1.1/api/_as_gen/matplotlib.pyplot.boxplot.html)
- [17] R. F. Kokaly, R. N. Clark, G. A. Swayze, K. E. Livo, T. M. Hoefen, N. C. Pearson, R. A. Wise, W. M. Benzel, H. A. Lowers, R. L. Driscoll, and A. J. Klein, *USGS Spectral Library Version 7: U.S. Geological Survey Data Series 1035*, 61 p., (USGS, 2017). Available at <https://crustal.usgs.gov/specclab/QueryAll107a.php>
- [18] Lawn\_Grass GDS91 green. USGS Spectral Library Version 7 (online, 2017). Available at [https://crustal.usgs.gov/specclab/data/HTMLmetadata/Lawn\\_Grass\\_GDS91\\_green\\_BECKa\\_AREF.html](https://crustal.usgs.gov/specclab/data/HTMLmetadata/Lawn_Grass_GDS91_green_BECKa_AREF.html)
- [19] *Grass\_Golden\_Dry GDS480*. USGS Spectral Library Version 7 (online, 2017). Available at [https://crustal.usgs.gov/specclab/data/HTMLmetadata/Grass\\_Golden\\_Dry\\_GDS480\\_ASDFRa\\_AREF.html](https://crustal.usgs.gov/specclab/data/HTMLmetadata/Grass_Golden_Dry_GDS480_ASDFRa_AREF.html)
- [20] Robert Šišma, Meteo Olomouc CZ. (online, 2020). Available at <http://infopocasi-olomouc.cz/>
- [21] Ondřej Vaculík, Multispectral image analysis, Project ID: 12176271, (online, 2020), GitLab repository. Available at <https://gitlab.com/vaculiko/multima>
- [22] Pedregosa *et al.*, *Scikit-learn: Machine Learning in Python*, (JMLR 12, pp. 2825-2830, 2011). Available at <http://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html>
- [23] P. Mishra, M. S. M. Asaari, A. Herrero-Langreo, S. Lohumi, B. Diezma, and P. Scheunders, *Close range hyperspectral imaging of plants: A review*, Biosystems Engineering, vol. 164, pp. 49-67, 2017.

# Contents of Enclosed DVD

data	
cube	Normalised MS images
fits	Calibrated FITS images
raw	Unprocessed images from the MS acquisition
solar	Data for solar irradiance plot
usgs	Selected USGS spectral library data
out	
ae	Autoencoder output – compressed data and models
clustering	Clustering output – labelled data and class spectral plots
indices	Calculated indices with visualisation and histograms
plots	Plots used in thesis text
0-preprocessing.py	Data normalisation script
0-statistics.py	Statistical data analysis
1a-clustering.py	Script for $k$ -means clustering
1b-autoencoder.py	Training and data compression script
indices.py	Index evaluation script
plots.py	Plotting script for figures used in this thesis
README.md	Detailed description of the data and script functions
utils.py	Functions used for plotting and data manipulation
DP-Vaculik-OPT-2020.pdf	Thesis text



# Appendices



Appendix 1: CMYK colour bars.



Appendix 2: Experimental setup inside the lab.

```

### Normalisation procedure
means = []
cubesN = []
for i, cube in enumerate(cubes):
    cubeNi = []
    Xnorm = slice(normareas[i][0][0], normareas[i][0][1])
    Ynorm = slice(normareas[i][1][0], normareas[i][1][1])
    for im in cube:
        imin, imax = (np.min(im), np.max(im[Xnorm, Ynorm]))
        im = (im - imin)*((newmax - newmin)/(imax - imin)) + newmin
        cubeNi.append(im)
        means.append(np.mean(im[Xnorm, Ynorm]))
    print(f'Cube {dates[i]} mean calculated.')
    cubesN.append(cubeNi)

mainmean = np.mean(means)

meansH = []
for i, cubeN in enumerate(cubesN):
    cubeH = []
    Xnorm = slice(normareas[i][0][0], normareas[i][0][1])
    Ynorm = slice(normareas[i][1][0], normareas[i][1][1])
    for imN in cubeN:
        imN += (mainmean - np.mean(imN[Xnorm, Ynorm]))
        cubeH.append(np.around(imN))
        meansH.append(np.mean(imN[Xnorm, Ynorm]))
    cubeOUT = np.ndarray.astype(np.asarray(cubeH), int)
    cubeOUT[cubeOUT > 65535] = 65535 # fix clipping values

```

Appendix 3: Core part of the image normalisation script 0-preprocessing.py.

```

### Databcube import and reshape
for date in files:
    name = date[:6]
    data = np.load(f'./data/cube/SEASONS/{date}')
    data = cropper(data, 800, 2000, 0, 2000)
    Path(f'./out/indices/').mkdir(parents=True, exist_ok=True)

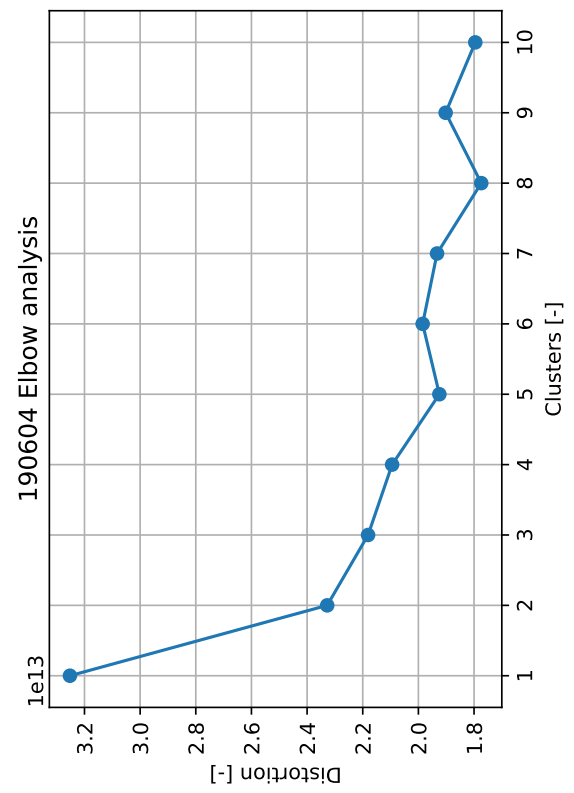
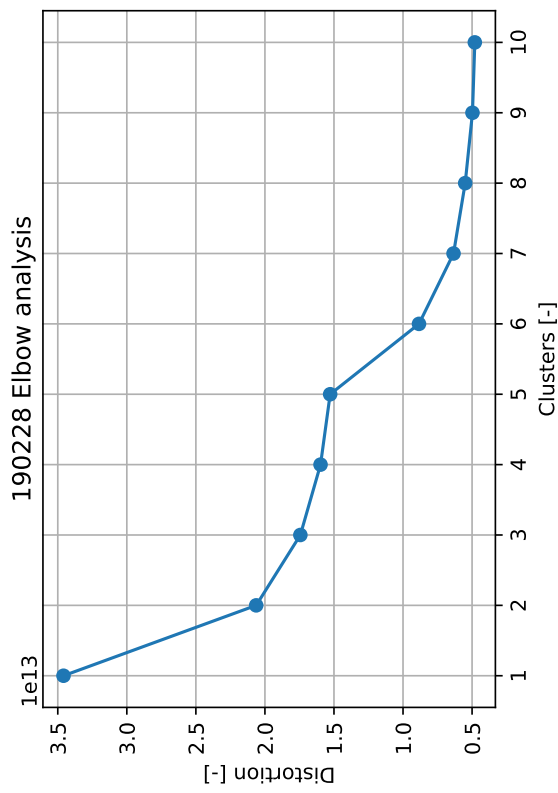
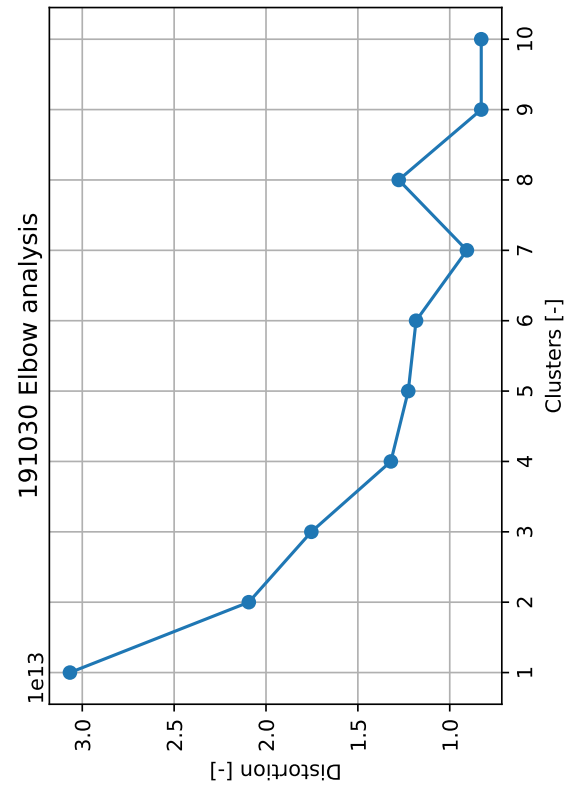
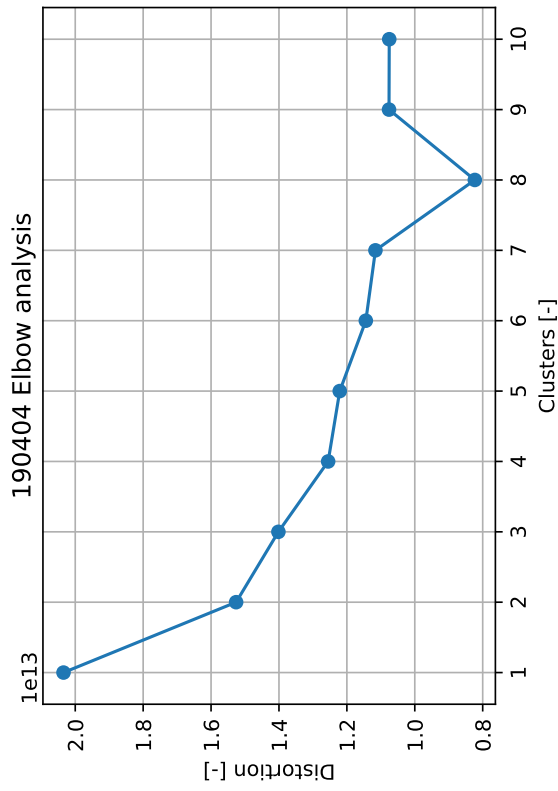
### Decompose cube to bands with filtername variables
fnames = ['f'+str(i) for i in filters]
for i, im in enumerate(data):
    vars()[fnames[i]] = im

### NDVI
ndvi = (f875-f650)/(f875+f650)
stats_ndvi = stats(ndvi)
imgshow(ndvi, f'{name} NDVI', save='png',
        folder='out\indices\plot')
imghist(ndvi, f'{name} NDVI', hrange=histrange(ndvi),
        save='png', folder='out\indices\plot')
np.save(f'./out/indices/{name}-NDVI', ndvi)

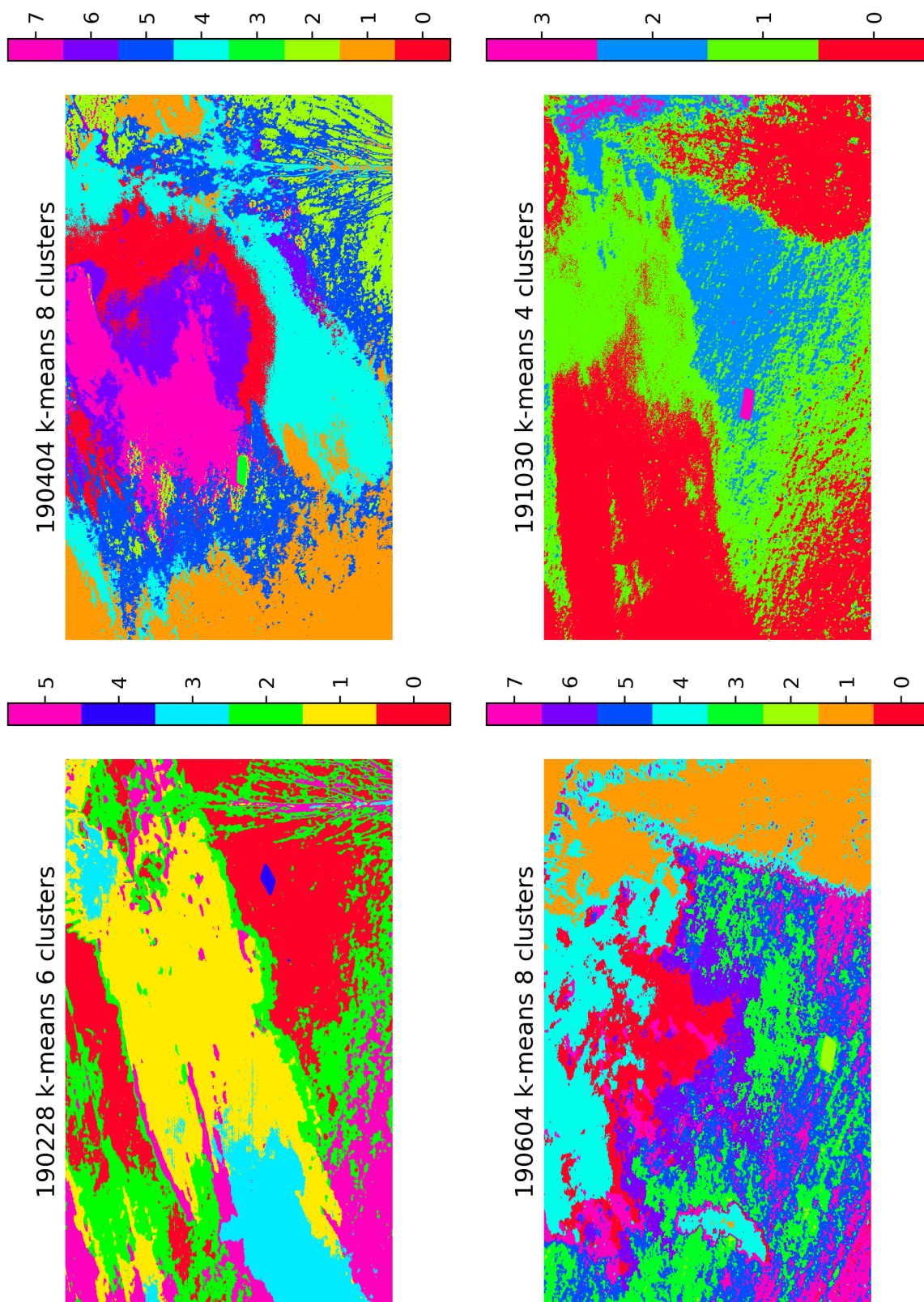
### Chorophyll index green
cig = (f875/f525)-1
stats_cig = stats(cig)
imgshow(cig, f'{name} CI-G', save='png',
        folder='out\indices\plot')
imghist(cig, f'{name} CI-G', hrange=histrange(cig),
        save='png', folder='out\indices\plot')
np.save(f'./out/indices/{name}-CI-G', cig)

```

Appendix 4: Core part of the index evaluation script `indices.py`.



Appendix 5: Elbow analysis for each measurement.



Appendix 6:  $k$ -means clustering output for each measurement.

```

### Autoencoder for dimensionality reduction
encoding_dim = 2
# Input spectrum data placeholder
input_spec = Input(shape=(6,))
# Encoding layer
encoded = Dense(encoding_dim, activation='sigmoid')(input_spec)
# Decoding layer
decoded = Dense(6, activation='sigmoid')(encoded)
# AE initialization
autoencoder = Model(input_spec, decoded)
encoder = Model(input_spec, encoded)
# Create a placeholder for an encoded (2-dimensional) input
encoded_input = Input(shape=(encoding_dim,))
# Retrieve the last layer of the autoencoder model
decoder_layer = autoencoder.layers[-1]
# Create the decoder model
decoder = Model(encoded_input, decoder_layer(encoded_input))
autoencoder.compile(optimizer='SGD', loss='mean_squared_error')
print("Autoencoder compiled.")

# Begin training
ae = autoencoder.fit(x_train, x_train,
                    epochs=epochs,
                    batch_size=batch_size, # in powers of 2
                    shuffle=True,
                    validation_data=(x_test, x_test))
print("Training ended.")

```

Appendix 7: Core part of the autoencoder training script 1b-autoencoder.py.