Pixel classification of hyperspectral images

Joakim Nyman
Abstract

Pixel classification of hyperspectral images

Joakim Nyman

For sugar producers, it is a major problem to detect contamination of sugar. Doing it manually would not be feasible because of the high demand and would require too much labor. This report evaluates if the problem can be solved by using a hyperspectral camera operating in a wavelength range of 400-1000 nm and a spectral resolution of 224. Using the machine learning algorithms Artificial Neural Network and Support Vector Machine, models were trained on pixels labeled as sugar or different materials of contamination. An autonomous system could be developed to analyze the sugar in real time and remove the contaminated sugar. This paper presents the results from using both Artificial Neural Networks as well as Support Vector Machine. It also addresses the impact of the pre-processing techniques filtering and maximum normalization when applying machine learning algorithms. The results showed that the accuracy can be significantly increased by using a hyperspectral camera instead of a normal camera, especially for plastic materials where using a normal camera gave a precision and recall score of 0.0 while using the hyperspectral camera gave above 0.9. Support Vector Machine performed slightly better than using Artificial Neural Network, especially for plastic material. The filtering and the maximum normalization did not increase the accuracy and could therefore be omitted in favor for performance.
Table of contents

Introduction ..................................................................................................................... 1
Problem statement ........................................................................................................... 1
Research methodology .................................................................................................. 2
Background .................................................................................................................... 3
Food waste and fraud ...................................................................................................... 3
Hyperspectral imaging .................................................................................................... 3
Machine learning ........................................................................................................... 4
  Support Vector Machine ............................................................................................... 5
  Artificial Neural Network ............................................................................................ 5
Related works .................................................................................................................. 8
Methods .......................................................................................................................... 12
Setup ............................................................................................................................... 12
Pre-processing ................................................................................................................ 13
  1. Reference normalization ......................................................................................... 13
  2. Savitzky-Golay filter .............................................................................................. 13
  3. Maximum normalization ........................................................................................ 13
Ground truth .................................................................................................................. 14
Training ........................................................................................................................... 14
Testing ............................................................................................................................. 16
Results ............................................................................................................................. 17
  1. RGB-image .............................................................................................................. 18
  2. SVM with Savitzky-Golay filter and maximum normalization .................................. 19
  3. ANN with Savitzky-Golay filter and maximum normalization ................................... 20
  4. SVM without Savitzky-Golay filter and no maximum normalization ....................... 21
Conclusion ...................................................................................................................... 22
Future work ..................................................................................................................... 23
References ....................................................................................................................... 24
Introduction

In the food industry quality checks are often performed manually such as: tenderness of meat, maturity of an avocado, or contamination. Sometimes they are not performed at all due to personnel cost or because they are simply too slow for the high demand. This report covers the specific case of foreign objects that have contaminated sugar. It presents the results of an investigation of how the detection of contamination can be automated by using a hyperspectral camera, and the machine learning techniques Support Vector Machine (SVM) and Artificial Neural Network (ANN). The foreign objects contaminating the sugar were of different materials such as: plastic, paper and metal. Experiments were made with the objects laying totally visible on the sugar surface but also with the objects hidden under the surface.

The introduction includes Problem statement where the main goal of this project is explained and Research methodology where the methods used to reach the goal are explained. After the introduction, a comprehensive Background section explains: the problem with food waste and fraud, the difference between hyperspectral images and normal images, the two machine learning techniques SVM and ANN, and ends with several examples of Related works made by other authors who have used hyperspectral cameras to solve different problems in the food industry. The camera used in these experiments was a hyperspectral line scanning camera operating in the wavelength range of 400 to 1000 nm, which includes wavelengths not visible for the human eye, and a spectral resolution of 224. Information about the camera and the other hardware used in the experiments is presented in the Setup section. Pre-processing of the data received from the camera was made by normalization and filtering which is detailed in the Pre-processing section. To find the foreign objects in the sugar machine learning algorithms were trained and later used on unseen data for evaluation. The ground truth data for training were created by manually labeling pixels and is explained in detail in the Ground truth section. An Artificial Neural Network with one hidden layer consisting of 10 neurons was tested as well as a linear Support Vector Machine. Details on how these were used are presented in the Training section. A qualitative study was made using only three spectral bands corresponding to red, green and blue colors for comparison against using all the bands and all the wavelengths achieved from the hyperspectral camera. Applying the filter in the pre-processing step showed to be very compute intensive. In production the throughput of the sugar is typically high, and the scanning application must be able to handle the speed. Further, since it is a scanning application only one line at a time is available which makes it infeasible to perform a maximum normalization against a whole image. Therefore, the Results section also includes the result of a model trained on data not pre-processed with the filter or the maximum normalization. This report ends with a discussion of the results in the Conclusion section and what could be investigated further in the Future work section.

Problem statement

A normal camera can be used in an application that automates the foreign object detection. The purpose of images captured by a normal camera, however, is to make them possible to be seen for the human eye. Therefore, they are captured using three different wavelengths of light in the visible spectrum. A machine, however, does not have the biological limitation of an eye and can therefore make use of wavelengths outside the visible spectrum. Further, a machine has the potential to process much more information than the human brain so that more than three wavelengths can be used. In practice, this should mean that a machine should be able to see the difference between objects which for the eye look the same. It should also mean that a machine can easier see the objects when they are hidden under the surface of, in this case, the sugar. The main goal of this
project was to find out how much the performance of foreign object detection can be improved by using hyperspectral imaging over using normal images.

**Research methodology**

Since most of the objects were already visible for the eye, at least when laying above the surface of the sugar, wavelengths in the visible spectrum were desirable. To also be able to find objects difficult to see for the eye, such as transparent plastic or objects hidden under the sugar surface, wavelengths outside the visible spectrum were desirable as well. Therefore, the hyperspectral camera used in this project operated in the whole visible spectrum as well as a part of the near infrared spectrum.

Two machine learning models were used in the evaluation. Support Vector Machine is a very common and widely used model because it has shown to be good at solving problems like the one in this project. Artificial Neural Networks have lately become very popular due to the success of deep learning. Hence, a comparison was made between these two machine learning models.

Hyperspectral images contain much more information than normal images which is good for the accuracy. That requires, however, more computations which makes the application slower but for a real-time application the speed may be crucial. Therefore, an experiment was made where a filtering process to smoothen the data was omitted to increase speed and hopefully not decrease the accuracy.
Background
This section starts by addressing the problem with **Food waste and fraud** with examples of what can be done to reduce it. The **Hyperspectral imaging** section explains hyperspectral images and the difference to normal images. The two different machine learning techniques SVM and ANN are explained in the **Machine learning** section. The last section **Related works** includes summaries on previous works that has been done using hyperspectral imaging in the food industry.

Food waste and fraud
According to the European Commission between 1/3 and 1/2 of the world food production is not consumed (1). Food waste is a big issue and connected to, e.g., climate changes, water, and availability of resources. It is also connected to economic such as resource efficiency, price volatility, increasing costs as well as social factors such as health and equality. This is enough food in the world and still many people suffer from hunger. The natural resources are decreasing while the world’s population is increasing. There are several actors in the food chain and they all must help reducing the food waste. Example of actors are the producers, e.g., farmers, manufacturers and processors, and actors who makes the food available for consumers, e.g., hospitality sector and retailers, and of course the consumers themselves. According to the European Commission half of the food waste comes from the consumers and more than one fourth of the food waste comes from the production and processors. One way of reducing the food waste at consumer level is to create awareness by reaching out with information about what they can do, e.g., using the freezer correctly, batch cooking, checking the dates (and knowing the difference between “use by” and “best before”). How to reduce the food waste at producer level can vary a lot dependent on what kind of product that is produced. It often consists of some kinds of quality checks, e.g., determine if the product is good enough to consume and what due date should be written on the label. Using a better quality check method, that can make a better due date prediction, would in turn reduce the food waste at consumer level. The quality checks could better decide if a product is good enough to be used for the purpose and if not, it could be used for something else, e.g., a steak that is too tough may be better suited as minced. Another important quality check is to detect if any foreign object has contaminated the product. Any foreign objects should be removed already at the producers to reduce the food waste later.

Another big issue in the food supply chain is food fraud. Food fraud can be, e.g., mixing different kinds of meat but market it as only one kind. During the years 2013 and 2014, official controls revealed that certain pre-packaged products contained horse meat even though this was not declared in the list of ingredients (2). It also happens that producers use medicinal products in the meat that are illegal to use in food-producing animals. Doing quality checks to reveal these kinds of frauds are time-consuming, labor-intensive and often require that the sample be destroyed and thus becomes useless.

Hyperspectral imaging
Hyperspectral images differ from common images by using a higher resolution in the spectral dimension and a wider wavelength range. The higher spectral resolution and wavelengths invisible for the human eye enables a computer to extract much more information from the captured object. In order to capture hyperspectral images, a hyperspectral imager is used which is a type of remote sensing instrument that quantifies electromagnetic radiation from a scene (3). The scene radiation comes from either solar energy or thermal energy or a combination of them both. Different types of hyperspectral imagers may sense and quantify different wavelengths of the electromagnetic radiation. Which wavelengths the hyperspectral imager can sense affects the kind of information one can get from the scene. Physical properties of the captured object affect how it reflects, absorbs and emits the different wavelengths of the radiation which affects the received information as well.
Figure 1 illustrates the difference between a hyperspectral image and a RGB (4) image. A RGB image is captured by using three different wavelengths, one for red which is around 610 nm, one for green around 550 nm and one for blue around 460 nm. The hyperspectral image to the left in the figure consists of more than three spectral channels in typically a much wider wavelength range. For example, the hyperspectral image could be captured using a wavelength range in 400-1000 nm with a resolution of 400 spectral channels.

![Figure 1: Hyperspectral image compared to RGB (4) image. RGB image typically has three spectral channels, red, green and blue, while a hyperspectral image can have several hundred spectral channels. (5)](image1)

The remote sensing instrument can be manufactured to sample a hyperspectral image in different ways as illustrated in Figure 2. Point-scanning spectrometer samples the spectral irradiance of a scene pixel per pixel, which can be combined to a three-dimensional dataset typically called a datacube. Line-scanning spectrometer samples the spectral irradiance line by line and a snapshot imaging spectrometer collect the entire three-dimensional datacube in a single integration period.

![Figure 2: (a) shows point-scanning and line-scanning spectrometer while (b) shows snapshot imaging spectrometer which samples the spectral irradiance of the entire scene in a single integration period. (6)](image2)

Since the development of imaging spectrometers in the 1980’s (6), NASA has been using them to detect different materials of celestial bodies (3). They have also been used to make ecological measurements such as quantity of different rock types or vegetation health as well as mapping characteristics of tropical rain forests, agricultural crops, forests, urban areas, water resources, geologic features, coral reefs etc. (7).

**Machine learning**

To extract useful information out of a data set one can make use of statistical models. Another way of extracting the information is by using algorithms that can learn from the data. This is called machine learning which is more common to use when dealing with higher dimensional data sets. When working with data in the form of images the most common type of problem to solve is classification. Since a RGB-image only consist of three spectral channels it makes sense to make the classification based on spatial information and a widely used machine learning model for that is convolutional neural networks (CNN). For hyperspectral images, however, that can have several
hundred spectral dimensions, it is more interesting to focus on the spectral dimensions rather than the spatial. A machine learning algorithm would instead learn to classify the pixels in the image independently. One critical problem to consider, however, is the large amount of data due to the large spectral dimensionality. As an example, one image of size 1024*1024 pixels would add 1 048 576 samples to the data set. Using 448 spectral channels, the raw data received from the camera would already be around 1 GB large for only one image. After pre-processing the size could be twice that size or even fourth that size depending on the arithmetic precision. Examples of traditional classification methods are k-nearest neighbors, maximum-likelihood, minimum distance, logistic regression etc. The two methods used in this report are Support Vector Machine and Artificial Neural Network which are explained in the rest of this section.

Support Vector Machine
The Support vector machine (SVM) is an old and popular machine learning method. It is not very different from an ANN in its simplest form, but for simpler problems it often turns out to perform better. It tries to find a set of parameters separating the classes. For a linear SVM, i.e. without a non-linear activation function, often called a kernel for SVM, the parameters define a hyperplane. It has the same model as a neural network with only one output neuron, without a non-linear activation function, and no hidden layers. Adding a non-linear kernel would be the same as adding a non-linear activation function in the neural network that projects the input into some other space. Since a SVM in its simplest form is a binary classifier, for more than two classes multiple SVMs must be trained separately. While a neural network is trained on all the training data iteratively by presenting one sample at a time, a SVM selects a sub-set of the training data to train on. The sub-set includes only samples that lies close to the other class. The parameters defining the hyperplane are found by maximizing a margin around it so that as many samples as possible are far away from the hyperplane. The way a SVM does this comes down to maximizing the squares of the distances. Since it is a quadratic function the surface to explore is convex, meaning it has no local minimums. In contrast to the gradient descent algorithm used for training a neural network which may lead to getting stuck in a local minimum, a SVM is guaranteed to find the optimal solution.

Artificial Neural Network
The use of Artificial Neural Networks (ANN) has heavily increased in popularity since deep learning started to show its usefulness, especially for computer vision. Further, in Hyperspectral imaging for food application (8), ANN is the proposed machine learning model for the classification problem in food applications, using hyperspectral imaging. The development of ANNs has been inspired by the operation of the biological central nervous system. An average human brain has about 100 billion neurons where each neuron may be connected to up to 10,000 other neurons (9). The connections are called synapses and there can be as many as 1 quadrillion of those. Every neuron in the brain maintains a voltage gradient and when the voltage changes significantly an electrical pulse is generated. The electrical pulses are transmitted to other neurons via the synapses and make an interaction between the neurons.

In an artificial neural network, neurons are created (often called nodes) and connected to each other through weighted connections. A value is computed based on the input to a neuron which in turn becomes the input to other neurons. Figure 3 shows a neuron with three inputs: x1, x2 and x3, which in analogue with a brain would be the outputs of three other neurons.
To calculate how much an output from a neuron affects a connected neuron the output value is multiplied with the weight of the connection between the two neurons. These weights are shown as: $w_1, w_2$ and $w_3$ in Figure 3, analogously the synapses in a brain. To calculate the total influence on the neuron, the inputs are multiplied with corresponding connection weights and are summed up according to the formula:

$$\text{weighted sum} = \sum_{i=1}^{n} w_i x_i$$

The computed dot product (weighted sum) of a neuron is used as either the input to another neuron or as the output of the whole neural network. Figure 4 shows an example of an artificial neural network consisting of multiple neurons connected to each other. The size of the input layer would be the same as the number of dimensions in the input data. Each input node is connected to every node in the next layer, usually called the hidden layer. Each node in the hidden layer is connected to every node in the output layer. For a classification problem, this neural network would be used to solve a problem with four different classes, one output node for each class.

Training a neural network is typically performed by adjusting the weights of the connections until the error of the classification is minimized. To calculate the error, the correct classification of the input data must be known, often called “ground truth”. The way for a machine learning model to learn by minimizing the error against some ground truth data is typically called “supervised learning”. The function that calculates the error is typically called a “cost function” and is chosen differently.
depending on the data and the problem to solve. The goal is to find values for the weights and the biases so that the cost function becomes 0. If the cost function is convex it has only one minimum and would converge relatively easy. For a more complicated problem, however, the cost function is typically not convex, it has multiple local minimums and the global minimum may be hard to find. To find the global minimum for such cost function there is popular algorithm called “gradient descent”. Intuitively one can think of the global minimum of the cost function as a deep valley and the local minimums as shallower valleys surrounding it. If a ball starts rolling down the slopes of the valleys, the goal is to get the ball rolling through the shallower valleys and end up in the deepest one. Choosing random values for the weights and the biases is to randomly drop the ball somewhere. The goal is to update the values so that the ball keeps rolling down the slope which can be done by computing the derivative of the cost function, resulting in something called the gradient vector. In every iteration, the values should be updated so that the derivative is negative to make sure that the cost function is converging to 0. A common problem when using gradient descent is to get stuck in a local minimum instead of finding the global one but there are several tricks that can be utilized to avoid this. Learning rate is a parameter used to set how much the weights and the biases should be changed when updated. A greater learning rate will make the ball roll further and might thus overshoot a local minimum. It will, however, make it more difficult for the ball to roll all the way down to the bottom of the global minimum. A smaller learning rate will slowly converge if in the global minimum but might get stuck in a local minimum before getting there. One trick is to change the learning rate during training, either manually or by applying a momentum to it. As long as the derivative of the cost function is negative, the learning rate is increased for faster convergence. When the derivative becomes positive it means that a minimum has been overshoot, step back and decrease the learning rate and try again.

A problem with using the derivative to minimize the error is that, since the computed dot product for a neuron is proportional to each of the inputs in a linear way, each partial derivative of it would be a constant. Meaning that, the gradient has no relationship to the input, and no changes would be made to the weights when the input changes. To solve this problem, a function must be applied to the dot product of each neuron making it nonlinear. This function is typically called an “activation function” and there exist several functions with different pros and cons for the purpose. One activation function often used is the sigmoid function:

\[
y = \frac{1}{1 + e^{-x}}
\]

![Figure 5: Sigmoid function which makes the output of a neuron in the range (0,1).](image)

One drawback with the sigmoid function, is that towards either end, the y values will respond less to changes in x, meaning that the gradient will be very small and eventually vanish. This problem is called “vanishing gradients” and the effect is that the network stops learning or learns very slowly.
Another commonly used activation function that does not have the vanishing gradient problem but is still nonlinear is “rectified linear unit” (ReLU):

\[ y = \max(0, x) \]

Figure 6: ReLU activation function which makes the output of a neuron in the range (0,∞).

It is linear for positive \( x \), but a negative weighted sum would become 0. One pro of ReLU is that the resulting network of weights may be less dense since neurons with a negative weighted sum will be 0. These neurons will stop responding to variations in error or input since the gradient is 0. This could, however, result in important neurons to be wrongly deactivated and make the learning more difficult.

Related works
D. Yang et al. used hyperspectral imaging to explore the feasibility to predict moisture content and storage time of cooked beef (11). 105 sliced beef samples were used and immediately packed and held in cold storage. At 5 different time points: 7, 10, 13, 16 and 19 days, 15 samples were scanned, and the moisture content was measured directly. By applying variable combination population analysis, they selected 10 optimal wavelengths between 400-1000 nm to use when scanning the samples using a line-scanning spectrometer. Because of the high number of dimensions, they applied principal component analysis and then applied the discrete cosine transform to the first three principal component images to end up with 30 textural features for one sample. A back-propagation neural network was trained with 70 of the samples and the remaining 35 were used for testing the trained model. Evaluation was made by calculating the R-square and the root mean square error (RMSE) values. Using both spectral and textural information, their prediction of the moisture content was 0.977 in R^2 and 0.9151 in RMSE which was considered a very good result.

Accurate labeling of products is difficult to verify and consumers heavily rely on it. A method that can verify that the labels are correct is therefore of great value to producers, retailers, and consumers. Crichton SOJ et al. explored the feasibility to perform binary classification for three different combinations of beef freshness: fresh and frozen-thawed, matured and matured frozen-thawed, as well as fresh and matured (12). They used a line-scanning spectrometer operating in the wavelength range 500-1010 nm with a spectral resolution of 323 dimensions. Each of the three datasets consisted of 192 samples (96 per beef condition) which were used to train a support vector machine with a split ratio of 3:1 between train and test set. Using all the 323 spectral components, a correct classification rate above 90% was achieved for all the three datasets. A comparison was made using the 10 most important spectral components but with not as good result, sometimes as low as only 50%.

Marshall S et al. explored different applications in hyperspectral imaging for food quality (8). The eating quality (taste) of baked sponges, vanilla and chocolate, was tested by skilled human tasters who assigned each sponge a score. The same sponges were tested repeatedly over time to quantify
the relationship between the eating quality and the age of a sponge until the taste was no longer acceptable, i.e., its shelf life. The hyperspectral imaging system used had a spectral range of 400-1000 nm. However, the eating quality of a sponge is correlated to the water content, i.e. the tasting score decreases when the sponge is drying out. Knowing that the spectral band at 970 nm is known to absorb water, that band alone is sufficient to find out the water content of a sponge, and thus an estimate of its eating quality can be made. The authors did not develop a system to predict the eating quality but the results from the measurements is shown in Figure 7. (a) and (b) shows that the increase in reflectance is more apparent around 970 nm than for any other band. (c) and (d) shows the relation between the scores from the testing panel and the calculated scores based on the reflectance using the spectral band at 970 nm.

![Vanilla Sponge Spectral Profile](image1)

![Chocolate Sponge Spectral Profile](image2)

**Figure 7:** (a) and (b) shows the relation between the reflectance of different wavelength bands and age of sponges. (c) and (d) shows the relation between the test panels scores and a calculated score from the reflectance of wavelength 970 nm.

Marshall S et al. also investigated the feasibility for classification of five types of Chinese tea using hyperspectral imaging. Tea is a highly consumed product at varying prices and quality between different brands. It needs to be verified for being the correct type and having the expected quality for the price. Manual classification of the type relies on subjective input from domain experts. Further, tea samples of different types appear almost identical to each other for the human eye. With a hyperspectral imaging system, 170 spectral bands were used in the wavelength range of 400-1000 nm. Dimensionality reduction was done by principal component analysis where the first 10
components were used. The authors trained an artificial neural network to do the classification. To increase the accuracy a filter of size 5x5 pixels was applied to define a correctly classified tea object. The confusion matrix in Figure 8 shows that hyperspectral imaging can be used to classify different objects even though they are visually similar.

<table>
<thead>
<tr>
<th>Actual</th>
<th>Predicted</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Tea 1</td>
</tr>
<tr>
<td>Tea 1</td>
<td>88.94%</td>
</tr>
<tr>
<td>Tea 2</td>
<td>0.20%</td>
</tr>
<tr>
<td>Tea 3</td>
<td>0.20%</td>
</tr>
<tr>
<td>Tea 4</td>
<td>1.67%</td>
</tr>
<tr>
<td>Tea 5</td>
<td>4.47%</td>
</tr>
</tbody>
</table>

Figure 8: Confusion matrix showing classification accuracy of five different types of tea.

Marshall S et al. made a comparison between using hyperspectral imaging in the wavelength range of 400-1000 nm and using conventional imaging systems, i.e. RGB-images where spatial information is used together with spectral. The subject of this study was four different types of rice with differences in shape, size, color and, assumingly, spectral response. Each image used for training contained 72 grains of the same type and each image used for testing contained 18 grains of each type. Five different datasets were produced based on: spatial information only; RGB-color information only; spectral information only; spatial and RGB-color information; and spatial and spectral information. A support vector machine was trained and used to evaluate the classification accuracy for each experiment. Figure 9 shows the classification accuracy results for the five different datasets. As the authors expected, the Spectral + Spatial dataset resulted in the best accuracy. What was surprising, however, was that Color + Spatial with 81.67% accuracy outperformed using only Spectral information which gave 74.27% accuracy. The authors showed that hyperspectral imaging can be useful when applied appropriately, and that together with standard image processing techniques, even better accuracy can be achieved.

<table>
<thead>
<tr>
<th>Data</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spatial</td>
<td>69.91%</td>
</tr>
<tr>
<td>Colour</td>
<td>67.17%</td>
</tr>
<tr>
<td>Spectral</td>
<td>74.27%</td>
</tr>
<tr>
<td>Colour + Spatial</td>
<td>81.67%</td>
</tr>
<tr>
<td>Spectral + Spatial</td>
<td>90.20%</td>
</tr>
</tbody>
</table>

Figure 9: Classification accuracy of four different types of rice using five different datasets based on different combinations of spatial information, RGB-color information and spectral information.

Zhu DS et al. did an experiment classifying different tea types (13). They collected 180 samples from six different famous types of Chinese tea, 30 for each type. 120 of the samples (20 for each type) were used for training and the remaining 60 samples were used for testing. A least squares-support vector machine was trained and used for the classification task. A line-scanning spectrometer operating in the VIS/NIR (visible and near-infrared, 400-1000 nm) range was used when capturing the images. Their experiment included a comparison between using only spectral information, only textural information and using both. PCA was applied and the first two principal component were
chosen. To create the textural information, a GLCM (gray-level co-occurrence matrix) were created from each of the two principal component images. Each matrix consisted of 12 textural variables resulting in 24 textural variables for each sample. Figure 10 shows the accuracies for the experiments and the results were excellent when combining spectral and textural information and using a least squares-support vector machine for classification.

<table>
<thead>
<tr>
<th>Tea type</th>
<th>Number</th>
<th>Spectra</th>
<th>GLCM</th>
<th>Spectra+GLCM</th>
</tr>
</thead>
<tbody>
<tr>
<td>GG tea</td>
<td>10</td>
<td>80</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>JG tea</td>
<td>10</td>
<td>100</td>
<td>60</td>
<td>100</td>
</tr>
<tr>
<td>LS tea</td>
<td>10</td>
<td>80</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>ML tea</td>
<td>10</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>MJ tea</td>
<td>10</td>
<td>100</td>
<td>80</td>
<td>100</td>
</tr>
<tr>
<td>LC tea</td>
<td>10</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>total</td>
<td>60</td>
<td>93.3%</td>
<td>90%</td>
<td>100%</td>
</tr>
</tbody>
</table>

Figure 10: Classification accuracy of six different tea types using only spectral information, only textural information and using both spectral and textural information.
Methods
This section starts by presenting the hardware used in the Setup section. Before applying the machine learning techniques, pre-processing must be made to the raw data received from the camera. There are totally three different steps and they are explained in order under the Preprocessing section. To train a machine learning model, the pixels must be labeled to create ground truth data. How this was done is explained in the Ground truth section. How the training data was created is crucial for the accuracy of the trained model. Further, deciding how the parameters are set for the SVM or the ANN may as well have a significant impact on the accuracy. This is explained in more detail in the Training section. Finally, in the Testing section, the testing data used for evaluating the different trained models is presented.

Setup
The setup for collecting the images consisted of a line scanning camera (1), two halogen lights (2), a moving tray (3), a computer (4) and the object to scan (5).

1. The camera was a line scanning spectrometer able to capture reflectance in the VNIR wavelength range (visible and near infrared, 400-1000 nm). The maximum spatial resolution for the camera was 1024 pixels for one line and a spectral resolution at 448 bands for each pixel. In the experiments for this project 512 pixels in spatial resolution and 224 bands in the spectral resolution was used.
2. To capture hyperspectral images a good light source is very important. Two light bars were used each containing three halogen lights. The lights were adjusted to point directly on the scanner line on the tray.
3. The moving tray was driven by a stepping motor that can be programmed and operated through serial communication.
4. The computer had a camera link PCI-card installed to communicate with the camera and to receive the scanned lines. The tray was operated through a USB-to-Serial converter and the images were scanned with a software provided by the camera manufacturer.
5. A white A4 size paper was put on the tray where the sugar and different objects were placed on to produce the images.

Figure 11: The setup used in the experiments consisting of: 1) camera, 2) lights, 3) tray, 4) computer, and 5) sugar with contamination.
Pre-processing
This section explains the 3 steps of pre-processing that was applied to the data before training a machine learning model. The Results section, however, also presents the results for when only applying the reference normalization and omits step 2 and 3.

1. Reference normalization
To make the images invariant to light and noise, a calibration step is needed, as suggested by the authors classifying different tea leaves (13). A dark and a bright image were created as references. The dark image was achieved by closing the shutter of the camera and capture 10 frames. From here on, a line that the camera scans will be called a frame and has the dimensions 512*224 (pixels*spectral bands). The bright image was achieved by scanning a white object for 10 frames as well. For each of the two reference images, a 2D mean vector, or mean frame, were created by taking the mean of all the 10 frames captured, meaning that the shape and the number of dimensions of the resulting dark and bright reference images were kept the same as for one frame. After capturing a sample frame (F), each value in it was normalized against the corresponding values in the dark and the bright reference frames according to this formula:

\[ F_{\text{normalized}} = \frac{F_{\text{sample}} - F_{\text{dark}}}{F_{\text{bright}} - F_{\text{dark}}} \]

2. Savitzky-Golay filter
To increase the signal-to-noise ratio, a Savitzky-Golay (14) (15) filter was applied on each pixel’s full spectrum. This filtering method iterates through each of the pixel’s reflectance values and applies a low-degree polynomial based on the neighboring reflectance values of that pixel. The open-source package SciPy (16) for Python provides a Savitzky-Golay filter ready to use. Inspired by the section about Savitzky-Golay filters in Numerical Recipes (14) a filter with a window length of 61, a polynomial order of 5, and no derivative was used. Figure 12 shows a graph of the 224 spectral values of a pixel in the range 400 to 1000 nm before the Savitzky-Golay filter was applied.

3. Maximum normalization
The values retrieved from the camera are in the range 0 – 4000 which is considered wide in the context of machine learning. Using Linear Scaling to Unit Range normalizes the data in the range 0-1 and may improve the performance of the classification model (17). The formula is the same as the one used in step 1 but the minimum value and the maximum value were replaced with the dark and the bright reference values. Some materials, however, e.g., foil, may reflect light more than the white object used for capturing the bright reference which makes some of the values greater than 1. Therefore, the last step of the pre-processing consisted of normalizing the whole image (I) to its maximum value pushing all the reflectance values down to 1 or below according to this formula:

\[ I_{\text{max normalized}} = \frac{I_{\text{smooth}}}{\max(I_{\text{smooth}})} \]

In comparison, Figure 12 shows the same pixel before and after the filter (step 2) and maximum normalization (step 3) had been applied.
Ground truth

Producing the ground truth data needed for supervised learning is often very time consuming. To produce the ground truth data, every pixel that is going to be used in the training must be labeled with the correct classification, e.g., sugar, plastic, paper. Scyven (18) is an application that is used to visualize and analyze hyperspectral images. It includes a tool that can be used to draw polygons on the image and put labels on them. All pixels inside a polygon will automatically be labeled with the label of that polygon. When finished drawing the polygons, the labeled pixels can be exported as a CSV-file containing the name of the file, labels, coordinates and all the spectral values. Since the CSV-file also contains the spectral values for the pixels, it can directly be used in training by extracting the labels and the spectral values.

Training

A presumption for being able to train a classification model is that there exist trends in the data. That is, data points labeled as one class should have a correlation while being separable from data points belonging to other classes. By plotting the pixels for different objects, one can visually determine the possibility to train a classification model. When working with hyperspectral images, optimizations can be made by reducing the number of spectral dimensions, resulting in both faster performance as well as better classification accuracy. By plotting pixels belonging to different classes, wavelengths that gives the same reflectance for different classes can be detected and removed since those wavelengths would only make it harder for the classification model to learn the difference between pixels belonging to different classes. As an example, Figure 13 shows an image containing sugar and one piece of white plastic. Two different pixels from the white plastic and two pixels from the sugar were randomly picked and their reflectance of the 224 wavelengths in the range 400 to 1000 nm is shown in Figure 14. It is visually clear that the trend at pixels belonging to the same class is similar while different from the trend of pixels belonging to a different class. The reflectance power might differ between pixels belonging to the same class but the important property here is that the trend is similar. From the graph in Figure 14, the most important wavelengths can be detected. For example, the reflectance of the first wavelengths around 400 nm is very similar for all the pixels, even between different classes. These wavelengths might be better to remove from the data and maybe start the range around 500 nm instead. Same for the reflectance of the higher wavelengths around 1000 nm. After analyzing the data like this, a wavelength range between 500 and 950 nm could probably be a good choice to use. Of course more pixels must be looked at before deciding which wavelengths to use, especially if the data contains more classes. In this project, however, the whole wavelength range from 400 to 1000 nm was used.
Figure 13: An image with sugar and a piece of white plastic. Four pixels have been chosen and their wavelengths are visualized in Figure 14.

Figure 14: The reflectance of the wavelengths in the range 400 to 1000 nm for four different pixels belonging to two different classes, white plastic and sugar as seen in Figure 13. The trend at the reflectance is similar for two pixels belonging to the same class but different to pixels belonging to a different class.

The training set was created out of 15 images where two images had all the objects totally visible. In the remaining 13 images, the objects were hidden under the surface of the sugar at different levels. After labelling, and throwing away boundary pixels with uncertain labels, the training set consisted of 950 163 pixels.

The neural network for classification was developed using the machine learning framework Tensorflow (19) in Python. Each pixel was a training sample and since each pixel consisted of 224 values, the number of inputs to the network was set to 224. The output layer consisted of 8 neurons, one for each class, with no activation function, meaning they are simply linear. ANNs have the advantage that one or more hidden layers can be added with an arbitrary number of neurons, making it easier to solve complex classification problems. In this project, models with both one and two hidden layers were tested and the number of neurons tested varied from 3 up to 200. The final model used, which gave good accuracy but still fast training times, consisted of one hidden layer with 10 neurons. To make the model non-linear, the ReLU activation function was applied to the hidden layers outputs. With an input size of 224 and 8 classes the neural network has \((224 \times 10 + 10) + (10 \times 8 + 8) = 2338\) parameters in total.

For training a SVM the scikit learn (20) framework for Python was used. To train a multiclass linear SVM, One-vs-rest was applied which results in one SVM per class. Each SVM solves a binary
classification distinguishing one class from all the other classes. To find out the predicted class of a sample, the outputs of the SVMs can be compared and the one with the largest value is the predicted class. For each SVM the output value is the distance from the sample to the hyperplane of that SVM and that is why the SVM with the largest value is chosen. Using the algorithm for solving the primal optimization problem results in a fixed size of parameters equal to the number of spectral bands plus one bias per SVM. For 8 classes the trained SVM model would have the same form as a neural network with 8 outputs, linear activations, and no hidden layers. Using 224 spectral bands the trained model would have $224 \times 8 + 8 = 1800$ parameters in total.

**Testing**

For testing the trained model, two test images were captured, one with the objects totally visible and one with the objects hidden under the surface. The ground truth data was created from the whole image only leaving out the borders between different classes to minimize the risk of mislabeling. The test images are shown in Figure 15 where the left image is the image with visible objects and the right image with the objects hidden. Figure 16 shows an image of the ground truth data created, it is color coded according to the pixels label. As when creating the ground truth data for training, boundary pixels with uncertain labels were omitted and are the white areas in Figure 16. Thus, the exported CSV-file would contain all the pixels except those colored in white. The two test images, together with the created CSV-file, were then used for testing the models that have been trained on the sugar and the objects in the training set.

![Figure 15: The left image is the original image that were used for testing the classification accuracy of visible objects. The right image is the one used for testing the accuracy of hidden objects.](image1)

![Figure 16: The ground truth data created from the two test images in Figure 15. Pixels in white areas were not used in the testing to create a safe margin between the different classes.](image2)
Results

The accuracy of a trained model was evaluated by using the model to classify all the pixels in the test images and compare the result to the ground truth data. Two accuracy numbers were calculated for each class using the commonly used metric Precision-Recall (21) (22). Both precision and recall consider how many pixels belonging to the class were classified positive. Precision also considers how many pixels not belonging to the class were classified positive. Precision, therefore, tells us how correct the model is at predicting a specific class. Recall, instead, considers the pixels only belonging to the class, meaning that how many pixels belonging to the class were classified positive and how many were classified negative. Recall, therefore, tells us how good the model is at finding the specific class.

\[
\text{Precision} = \frac{\text{True positive}}{\text{True positive} + \text{False positive}}
\]

\[
\text{Recall} = \frac{\text{True positive}}{\text{True positive} + \text{False negative}}
\]

This section continues with showing the results of these four different experiments:

1. To make a qualitative comparison the section RGB-image shows the result for where an SVM was trained on only three bands, 460, 527, and 658 nm, to simulate a non-hyperspectral camera. The Savitzky-Golay filter was not applied here since the data had only three dimensions per pixel.

2. In the section SVM with Savitzky-Golay filter and maximum normalization an SVM was trained on all the 224 spectral bands in the range 400 to 1000 nm. The Savitzky-Golay filter was applied to the data followed by the maximum normalization.

3. In the section ANN with Savitzky-Golay filter and maximum normalization an ANN was trained on all the spectral bands and the full wavelength range, with both the filter and the maximum normalization applied.

4. In the section SVM without Savitzky-Golay filter and no maximum normalization an SVM was trained on data where the filter and the maximum normalization steps were not applied. It was evaluated as a comparison to the second experiment to find out if the two pre-processing steps are really needed or if they can be omitted in favor for speed.

Each experiment result is presented by first showing a color-coded image based on the classification of the test image with visible objects, and then one for the test image with hidden objects. Next to each image is a table showing the recall and precision scores for each class. The colors in the first column shows which color each class is color-coded as in the images.
1. RGB-image

The SVM model using RGB-images was bad at separating the different materials, e.g., brown paper and bristle. The bristle was mostly classified as brown paper, hence the low recall of 0.14. Same for the clip which was detected as an object but not as a clip, hence 0.0 in both recall and precision. Neither the white plastic nor the cable tie was found at all, hence 0.0 in both recall and precision.

![Image](image1.png)

**Figure 17**: SVM model trained on three RGB-bands. The image is the classification result for the test image containing visible objects.

When the objects were hidden under the sugar, the accuracy became much worse. The classifier could still detect the foil well, probably because of the high reflectance of the material. The precision for sugar at 0.93 may seem good but is misleading. The sugar class contained many more samples compared to the other classes and therefore it is actually a very bad score for being the sugar class.

![Image](image2.png)

**Figure 18**: SVM model trained on three RGB-bands. The image is the classification result for the test image containing hidden objects.

<table>
<thead>
<tr>
<th>Predicted class</th>
<th>Recall</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clip</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Black plastic</td>
<td>0.71</td>
<td>0.64</td>
</tr>
<tr>
<td>White plastic</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Bristle</td>
<td>0.14</td>
<td>0.99</td>
</tr>
<tr>
<td>Brown paper</td>
<td>1.0</td>
<td>0.81</td>
</tr>
<tr>
<td>Foil</td>
<td>0.68</td>
<td>0.84</td>
</tr>
<tr>
<td>Transparent cable tie</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Sugar (not object)</td>
<td>1.0</td>
<td>0.98</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Predicted class</th>
<th>Recall</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clip</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Black plastic</td>
<td>0.0</td>
<td>0.08</td>
</tr>
<tr>
<td>White plastic</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Bristle</td>
<td>0.01</td>
<td>0.6</td>
</tr>
<tr>
<td>Brown paper</td>
<td>0.02</td>
<td>0.45</td>
</tr>
<tr>
<td>Foil</td>
<td>0.22</td>
<td>0.81</td>
</tr>
<tr>
<td>Transparent cable tie</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Sugar (not object)</td>
<td>1.0</td>
<td>0.93</td>
</tr>
</tbody>
</table>
2. SVM with Savitzky-Golay filter and maximum normalization

For SVM using hyperspectral images the separation of the classes worked very well. The sugar had perfect scores of 1.0 and even the white plastic and the cable tie were both found with scores above 0.9.

![Figure 19: SVM model trained on 224 spectral bands in the range 400 – 1000 nm. Savitzky-Golay filter was applied and all pixel values were normalized to the maximum value. The image is the classification result for the test image containing visible objects.](image19)

Even with the objects hidden under the sugar, the classifier succeeded in finding the objects. The black plastic, however, got mixed up with other objects, hence the low recall of 0.06. This was mostly because of the brown paper which can be seen on the precision score dropping to 0.86 compared to 0.98 when the objects were visible. The precision for the bristle dropped to 0.26 which can be seen in the image being due to pixels surrounding the bristle being wrongly classified.

![Figure 20: SVM model trained on 224 spectral bands in the range 400 – 1000 nm. Savitzky-Golay filter was applied and all pixel values were normalized to the maximum value. The image is the classification result for the test image containing hidden objects.](image20)

<table>
<thead>
<tr>
<th>Predicted class</th>
<th>Recall</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clip</td>
<td>0.73</td>
<td>0.99</td>
</tr>
<tr>
<td>Black plastic</td>
<td>0.95</td>
<td>0.95</td>
</tr>
<tr>
<td>White plastic</td>
<td>0.92</td>
<td>0.97</td>
</tr>
<tr>
<td>Bristle</td>
<td>1.0</td>
<td>0.76</td>
</tr>
<tr>
<td>Brown paper</td>
<td>1.0</td>
<td>0.98</td>
</tr>
<tr>
<td>Foil</td>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td>Transparent cable tie</td>
<td>0.95</td>
<td>0.97</td>
</tr>
<tr>
<td>Sugar (not object)</td>
<td>1.0</td>
<td>1.0</td>
</tr>
</tbody>
</table>
3. ANN with Savitzky-Golay filter and maximum normalization

Training an ANN with hyperspectral images performed almost as well as the SVM. The sugar, however, dropped to 0.99 in recall which means that some pixels belonging to the sugar class were classified as an object. Looking at the image those pixels are mostly black which is the color of white plastic. This is consistent with that the precision for white plastic dropped to 0.66 from 0.99 when using SVM.

![Image](image1.png)

Figure 21: ANN model trained on 224 spectral bands in the range 400 – 1000 nm. Savitzky-Golay filter was applied and all pixel values were normalized to the maximum value. The image is the classification result for the test image containing visible objects.

For the image with hidden objects the problem mixing the sugar with the white plastic became even more substantial. The recall for sugar dropped to 0.93 and the precision for the white plastic dropped to 0.13.

![Image](image2.png)

Figure 22: ANN model trained on 224 spectral bands in the range 400 – 1000 nm. Savitzky-Golay filter was applied and all pixel values were normalized to the maximum value. The image is the classification result for the test image containing hidden objects.
4. SVM without Savitzky-Golay filter and no maximum normalization

As explained in the Introduction, applying the Savitzky-Golay filter to every pixel is compute intensive and it may be hard to get the high throughput needed. Further, since only one frame at a time is available in a real-time application, the Maximum normalization is not possible which requires access to a whole image. In this experiment both were excluded from the pre-processing, meaning that only the Reference normalization was applied to each frame before training and testing a model. The model used in this experiment was SVM, simply because it performed best in the previous experiments. These results are therefore compared to the results in SVM with Savitzky-Golay filter and maximum normalization. The results when omitting step 2 and 3 in the pre-processing were very similar to the results including all the pre-processing steps. The most protruding difference was that bristle dropped from 0.76 to 0.37 in precision. Further, the recall for sugar dropped from 1.0 to 0.99 which can be seen in the image where some pixels in the sugar class were classified as objects.

![Figure 23: Savitzky-Golay filter was not applied and the pixels were not normalized against the maximum value. The image is the classification result the test image containing visible objects.](image)

<table>
<thead>
<tr>
<th>Predicted class</th>
<th>Recall</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clip</td>
<td>0.85</td>
<td>0.95</td>
</tr>
<tr>
<td>Black plastic</td>
<td>0.93</td>
<td>0.95</td>
</tr>
<tr>
<td>White plastic</td>
<td>0.95</td>
<td>0.82</td>
</tr>
<tr>
<td>Bristle</td>
<td>1.0</td>
<td>0.37</td>
</tr>
<tr>
<td>Brown paper</td>
<td>1.0</td>
<td>0.98</td>
</tr>
<tr>
<td>Foil</td>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td>Transparent cable tie</td>
<td>0.99</td>
<td>0.9</td>
</tr>
<tr>
<td>Sugar (not object)</td>
<td>0.99</td>
<td>1.0</td>
</tr>
</tbody>
</table>

For the image with hidden objects, however, the same numbers that dropped when using the image with visible objects were now increased. The precision for bristle increased from 0.26 to 0.39 and the recall for sugar increased from 0.98 to 0.99.

![Figure 24: Savitzky-Golay filter was not applied and the pixels were not normalized against the maximum value. The image is the classification result the test image containing hidden objects.](image)

<table>
<thead>
<tr>
<th>Predicted class</th>
<th>Recall</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clip</td>
<td>0.51</td>
<td>0.84</td>
</tr>
<tr>
<td>Black plastic</td>
<td>0.07</td>
<td>0.72</td>
</tr>
<tr>
<td>White plastic</td>
<td>0.58</td>
<td>0.49</td>
</tr>
<tr>
<td>Bristle</td>
<td>1.0</td>
<td>0.39</td>
</tr>
<tr>
<td>Brown paper</td>
<td>0.99</td>
<td>0.88</td>
</tr>
<tr>
<td>Foil</td>
<td>0.99</td>
<td>0.99</td>
</tr>
<tr>
<td>Transparent cable tie</td>
<td>0.88</td>
<td>0.78</td>
</tr>
<tr>
<td>Sugar (not object)</td>
<td>0.99</td>
<td>0.99</td>
</tr>
</tbody>
</table>
Conclusion

This report investigated how a camera operating in a wider wavelength range, \(400 - 1000\) nm, and with 224 spectral dimensions instead of only three, as used in normal cameras capturing RGB-images, can improve image analysis. The investigation was made by using the machine learning techniques Artificial Neural Network (ANN) and Support Vector Machine (SVM) to classify the pixels in the images. The images consisted of sugar and 7 objects of different materials, both visible and hidden under the sugar. The objects were clip, black plastic, white plastic, bristle, brown paper, foil, and transparent cable tie.

The result of four different experiments were presented. One where a SVM was trained using only the three RGB spectral bands to make a qualitative comparison with using the 224 spectral bands achieved from the hyperspectral images. For the second experiment all the 224 spectral bands were used to train a SVM. For the third experiment, all the 224 spectral bands were used but to train a ANN instead. The pre-processing consisted of three steps where the second step consisted of applying a Savitzky-Golay filter and the third step of normalizing all the values in an image to the maximum value of it. Applying the filter is compute intensive and may not be feasible to perform in a high-speed application. Further, since the camera was a line-scanner the third step would not be possible to perform since only one line at a time is available and not a whole image. Therefore, a forth experiment was made where these two pre-processing steps were omitted to compare with the result of the second experiment.

Each of the four experiments were evaluated by testing the trained model on two images, one with the objects laying visible on the surface of the sugar and one with the objects hidden under the sugar. On the image with visible objects the RGB-model did quite well separating the sugar from object with a recall of 1.0 and a precision of 0.98. However, it did not find the white plastic and the cable tie which both had 0.0 in recall and precision. The model also had problem with separating different objects from each other, e.g., it classified the bristle as brown paper. The model did not well on the image with the objects hidden under the sugar surface. While the recall score of the sugar was at 1.0 the precision score was only 0.93. The precision score might seem to be not that bad but considering that the sugar class contains much more samples than the objects, 1.0 in recall and 0.93 in precision means that most of the objects were classified as sugar.

The second model, when using all the spectral bands, performed very well on separating the objects from the sugar with both the recall and the precision scores at 1.0 for the image with visible objects. Where this model really outperformed the RGB-model was for, the white plastic with a recall of 0.92 and a precision of 0.97, and the transparent cable tie with a recall of 0.95 and a precision of 0.97. This model also outperformed the RGB-model when classifying the image with hidden objects. It separated the objects from the sugar with a recall of 0.98 and a precision of 0.99. It found the white plastic with a recall of 0.67 and a precision of 0.44, and the cable tie with a recall of 0.92 and a precision of 0.67. This is a huge improvement over the model trained on RGB-images which had recall and precision scores of 0.0 for both.

The third model trained used a ANN to compare with using a support vector machine and it performed almost as well. While the SVM had 1.0 in recall for the sugar the ANN had 0.99 for the image with visible objects. For the image with hidden objects the recall for the sugar dropped to 0.93 compared to the SVM with 0.98. Looking at the right image in the section ANN with Savitzky-Golay filter and maximum normalization, it contains a lot of noise in the sugar class. Most of the noise was due to the white plastic with a recall of 0.84 but a precision of only 0.13.
So far it has been shown that using hyperspectral images can give a huge improvement in accuracy over using normal RGB-images. To use a trained classifier in a real application, e.g., in a sugar warehouse, it must work in real time. The throughput in production is typically high which means that the computations must be fast. Applying the Savitsky-Golay filter turned out to be compute-intensive which makes it hard to meet the speed requirements. Further, the maximum normalization requires access to a whole image but since it is a line-scanning application only on line is available at a time. This means that the normalization against the maximum value of a whole image made in the earlier experiments would not be possible to make. Therefore, the fourth and last experiment investigated if the filter and the maximum normalization is necessary to get a decent result. Since SVM turned out to perform slightly better than ANN in the previous experiments, it was chosen as the algorithm to use in this experiment. Compared to the SVM model trained on data pre-processed with the filter and the maximum normalization the results were very similar. The experiment without the filter and the maximum normalization performed slightly worst on the image with visible objects but slightly better on the image with hidden objects. Therefore, the filtering and the maximum normalization steps can be omitted if the application is of line-scanning type and the speed is crucial.

Future work

Different materials reflect and absorb light differently for different wavelengths. Let’s say the problem to solve is to determine the freshness of an apple by finding out how much water it contains. Then, the wavelengths for which water reflects the light most should be used. Using other wavelengths would unnecessarily increase the number of dimensions in the data and could make the training more difficult. Translated into the problem of classifying different materials, an analysis should be made on the different wavelengths to see which ones differ most for the different materials. This can be done by, for each wavelength, calculate the variance between pixels belonging to different classes, and choose the wavelengths for which the variance is greatest. The authors in (13) uses Principal Component Analysis to find the optimal wavelengths to use for classification of tea leaves.

In the experiments it was shown that white plastic and especially transparent plastic were hard to distinguish from the sugar. The camera used in the experiments could only operate in the wavelength range of 400 – 1000 nm. Using a camera that could operate on wavelengths outside that range might make it easier to distinguish the plastic from the sugar. The light might also be better at penetrating materials, making it easier to detect objects that are hidden deeper below the sugar surface. For example, infrared light operates in the wavelength range of 700 nm - 1 mm, and ultraviolet in the range of 10 nm – 400 nm. Or thermal-infrared in the range of 3 um – 20 um, which is used by the military for camouflage detection since warm bodies radiate in that range.

In machine learning there are many different approaches and methods developed to solve different kinds of problems. A popular and widely used artificial neural network model called Convolutional Neural Network (CNN) are often the first choice when the goal is to find objects in images and classify them. The data for these image classification problems, however, typically consists of regular RGB-images and requires access to the whole image beforehand. In contrast, this project used images with many spectral dimensions. The need of the whole image beforehand would be a problem though, since the camera in this project was a line scanner used to continuously scan sugar and alarm as soon as an object is found. A solution would be to collect a fixed number of frames and then do the classification of an image composed by the collected frames. Another problem with CNN is that it does not natively support finding the spatial location of the objects found, which is directly available when doing the classification pixel by pixel as in this project. Finding the spatial location of objects when using CNN could be achieved by, in addition to the object classes, also label the images
with coordinates of each object's location, e.g., \((x, y, \text{width, height})\). An additional output layer would have to be added to perform regression on the four numbers representing the coordinates.

References


