Hyperspectral Imaging and Deep Learning for Food Safety Assessment

Mahmoud Yaseen Mubarak Al-Sarayreh

A Thesis Submitted to Auckland University of Technology
in Fulfilment of the Requirements of the Degree of
Doctor of Philosophy

School of Engineering, Computer and Mathematical Sciences
Auckland University of Technology
New Zealand

2020

Declaration

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the qualification of any other degree or diploma of a university or other institution of higher learning.

Signature of candidate

Dedication

To my wife Aysar, my parents Jawaher and Yaseen, my sisters and brothers, and my little daughter (Elaine). It is all for you.

Abstract

Hyperspectral imaging (HSI) systems are valuable tools for merging both spectroscopic and computer vision technologies in a single system with the advantage of providing the physical attributes and chemical distribution of materials represented in an image. Snapshot HSI systems are portable systems enabling the generation of HSI images at the video rate with limited spectral information, which makes this technology a step toward real-time HSI applications. For food processing, HSI systems are considered to form a rapid, non-invasive, non-destructive and chemical-free technology for predicting food attributes regarding food safety and quality assessment, which reflects positively on costs, accuracy and processing time of applications in the food industry.

This thesis studies the interaction between chemical and textural distributions, presented as spectral and spatial features of HSI images. Moreover, the thesis discusses several traditional and novel approaches in computer vision and deep learning for utilising this interaction in applications of safety assessment of food such as adulteration detection in meat products, authenticity of meat products and foreign object detection (FOD) in meat products.

In application of adulteration detection, traditional approaches are investigated for detecting the adulteration in meat products including spectral features and handcrafted textural features obtained from HSI images. Moreover, this thesis presents a novel multi-structure deep learning model for self-feature extraction and combination of these distributions (i.e., chemical and textural) in a single prediction model by using convolution neural networks (CNN). The model is evaluated against the traditional approaches and showed efficiency in prediction.

Extraction of joint spectral and spatial features from HSI images is also discussed in this thesis. A 3D–CNN approach is proposed for extracting the joint features. Red-meat classification (case study of fine-grain material classification) is used for evaluating the proposed approach. Moreover, we propose a novel graph-based postprocessing method for enhancing the prediction of the 3D–CNN approach or of any pixel-wise classification model. The proposed classification framework is evaluated against traditional machine learning algorithms such as support vector machines and partial least square discriminant analysis. Three datasets were collected for the evaluation by using three HSI systems: line scanning, near-infrared (NIR) snapshot and visible (VIS) snapshot HSI.

In application of FOD in meat products, the thesis discusses the object detection problem in HSI images based on their spectral and spatial features. A novel sequential deep learning framework is proposed for foreign object localization and classification by using CNN networks. The framework includes three modules in a sequential flow: Region proposal, filtering and classification modules. Two in-

dependent datasets of NIR snapshot HSI images, contaminated by many types of foreign materials, were used for training and testing the proposed approach. The evaluation showed promising efficiency of the proposed framework in terms of accuracy and real-time processing, compared with a baseline method for FOD such as the selective search approach.

Keywords: Food processing; Meat processing; Adulteration detection; Meat authenticity; Hyperspectral imaging; HSI; Snapshot HSI; Material classification; Spectral; Spatial; Textural; Joint features; Object detection; Foreign object; FOD; Foreign bodies; Deep learning; 2D-CNN; 3D–CNN; Region proposal; RPN; Classification

Acknowledgements

Words cannot express my heartfelt gratitude to my family. I am greatly thankful to my beloved parents, my wonderful wife, and my sisters and brothers, who have always believed in my ability to complete this research and encouraged me even through the most difficult stages with their ever-present love, support and prayers. Without their generous help and encouragement, this PhD thesis would not have been possible.

I owe a huge debt of gratitude to my main supervisor, Professor Reinhard Klette; without his invaluable guidance and encouragement, this thesis would never have been accomplished, not even started. Also, I thank co-supervisor Associate Professor Dr. WeiQi Yan for his support during this journey.

I am deeply grateful to my co-supervisor Dr. Marlon Reis for his great support, help, and guidance during the past three years. His support and invaluable inspiration helped me a lot at various stages of my research.

I am also thankful to the Auckland University of Technology (AUT) and AgResearch (through the Strategic Science Investment Fund) for the scholarship provided, and for providing the equipment for the development of research reported in this thesis. Their countless support and outstanding environments are highly appreciated.

Lastly, I thank all my colleagues at AUT's research centre CeRV and AgResearch's Meat Quality research team. I would also like to extend thanks to Dr. Mariza Reis for her kind support and encouragement. My deepest appreciations go to my colleagues and friends in New Zealand and Jordan for their support and always being there for me.

Mahmoud Yaseen Al-Sarayreh
Auckland, New Zealand
July 2020

Co-authored Publications

Outcomes of research, performed during the course of my PhD programme (reported in this thesis only to some degree), have been disseminated in refereed international journals, conferences, and at significant events. The following list also contains two submitted manuscripts:

Peer-reviewed journal papers:

- Reis, M., Van Beers, R., **Al-Sarayreh, M.**, Shorten, P., Qi Yan, W., Saeys, W., Klette, R., Craigie, C.: Chemometrics and hyperspectral imaging applied to assessment of chemical, textural and structural characteristics of meat. *Meat Science*, 2018, 144: 100-109.

- **Al-Sarayreh, M.**, Reis, M., Qi Yan, W., Klette, R.: Detection of red-meat adulteration by deep spectral–spatial features in hyperspectral images. *Journal of Imaging*, 2018, 4, 63.

- **Al-Sarayreh, M.**, Reis, M., Qi Yan, W., Klette, R.: Potential of deep learning and snapshot hyperspectral imaging for classification of species in meat. *Food Control*, 2020, 117, 107332.

Peer-reviewed conference papers:

- **Al-Sarayreh, M.**, Reis, M., Qi Yan, W., Klette, R.: Detection of adulteration in red meat species using hyperspectral imaging. *In Proc. of the Pacific-Rim Symposium on Image and Video Technology*, 2017, Springer.

- **Al-Sarayreh, M.**, Reis, M., Qi Yan, W., Klette, R.: Deep spectral-spatial features of snapshot hyperspectral images for red-meat classification. *In Proc. of the International Conference on Image and Vision Computing New Zealand*, 2018, IEEE.

- **Al-Sarayreh, M.**, Reis, M., Qi Yan, W., Klette, R: A sequential CNN approach for foreign object detection in hyperspectral images. *In Proc. of the International Conference on Computer Analysis of Images and Patterns*, 2019, Accepted but not yet published.

- **Al-Sarayreh, M.**, Moayed, Z., Bollard-Breen, B., Ramond, J. B., Klette, R.: Detection and spatial analysis of fairy circles. *In Proc. of the International Conference on Image and Vision Computing New Zealand*, 2016, IEEE.

- Rapson, C., Boon-Chong Seet, B, Lee, K., Naeem, N., **Al-Sarayreh, M.**, Klette, R.: Reducing the pain: A novel tool for efficient ground-truth labelling in images. *In Proc. of the International Conference on Image and Vision Computing New Zealand*, 2018, IEEE.

Conference and event presentations:

- **Al-Sarayreh, M.**, Reis, M., Qi Yan, W., Klette, R.: Red meat discrimination using hyperspectral imaging. NZIFST Conference, Nelson, New Zealand, 4 – 6 July 2017 (poster presentation).
- **Al-Sarayreh, M.**, Reis, M., Qi Yan, W., Klette, R.: Texture features in hyperspectral imaging. The 18th ANISG/NZNIRSS Conference, Rotorua, New Zealand, 11 – 12 April 2018 (oral presentation).
- **Al-Sarayreh, M.**, Reis, M., Qi Yan, W., Klette, R.: A deep learning approach for detecting the adulteration in red-meat products by hyperspectral imaging. IEEE Annual Workshop on Smart Sensors, Measurements and Instrumentation for Health, Food, Agriculture, Environment and Security. Palmerston North, New Zealand, 6 – 7 September 2018 (oral presentation).

Contents

List of Figures

List of Tables

List of Symbols

Symbols	Description
H	3D image (calibrated HSI image in reflectance)
R	3D image (raw HSI image in irradiance)
W	3D image (white reference HSI image in irradiance)
D	3D image (dark reference HSI image in irradiance)
D_x, D_y, D_λ	3D images domains; spatial and spectral
x, y, λ	Real variables; spatial location (x, y) and wavelength in spectral domain
N_{col}, N_{rows}	Number of columns and rows in an HSI image
N_{bands}	Number of wavelengths in an HSI image
w, h, d	Size parameters (width, height, and depth)
v	Value in a space
i, j, k, l, m, n	Natural numbers
f	Function
$\mathcal{L}$	Loss function
θ	Model parameters
W	Weight matrix
$\mathbf{b}$	Bias vector
Y, y	Ground-truth and predicted class label
n	Number of samples
C	Number of classes
K	Kernel of an CNN layer
M	Number of feature maps
s	Stride size
W_s	Spatial window size
SP	Set of superpixel labels
SN	Set of superpixel labels of neighbours of an superpixel
S	Total number of superpixels
N	Number of neighbours of an superpixel
w_t, w_n	weight factors of target superpixel and its neighbours
P_i	Model output of i-th superpixel
PN_j	Model output of j-th neighbour of i-th superpixel
ℓ	Tunable term of the penalization term
μ, σ	Mean value and standard deviation
Ω	Image carrier, set of all $N_{col} \times N_{rows}$ pixel location
$\mathbb{R}$	Set of real numbers
Σ	Summation
$\in$	Belonging to a set

List of Abbreviations

ANN *artificial neural networks*. 41
ANOVA *analysis of variance*. 37
AP *Average precision*. 156

CLSTM *convolutional long short-term memory*. 46
CNN *Convolution neural networks*. 46
CT *Computed tomography*. 6

DBN *deep belief networks*. 46

FNR *false negative rate*. 161
FOD *foreign object detection*. 133
FPR *false positive rate*. 161, 162
fps *frame per second*. 9

GLCM *gray level co-occurrence matrix*. 40
GT *ground truth*. 57

HSI Hyperspectral imaging. 5

IoU *intersection over union*. 139

KNN *k-nearest neighbour*. 41
KS *Kennard stones*. 62

LD *longissimus dorsi*. 35
LDA *linear discrimination analysis*. 41
LOOCV *leave one out cross validation*. 45
LR *logistic regression*. 44

MSC *multiplicative scatter correction*. 31
MSS *mean shift segmentation*. 39

NIR *near infra-red*. 7
NMRI *nuclear magnetic resonance imaging*. 134
NMS *non-maximum suppression*. 139

PCA *Principal components analysis*. 34
PLS *partial least squares*. 36
PLS–DA *partial least square discriminant analysis*. 41
PM *psoas major*. 35

RBF *radial basis function*. 45
ReLU *rectified linear units*. 72
RFE *recursive feature elimination*. 37
RNN *recurrent neural networks*. 46
ROI *region of interest*. 35
RPN *region-proposal networks*. 133
RR *recall rate*. 154

SAE *stacked autoencoder*. 46
SGD *stochastic gradient descent*. 73

SIMCA *soft independent modelling by class analogy*. 43
SLIC *Simple linear iterative clustering*. 55
SNV *standard normal variate*. 31
SPA *successive projections algorithm*. 37
SSD *single shot Multi-Box detector*. 136
ST *semitendinosus*. 35
SVM *support vector machines*. 41
SWIR *Short-wave infra-red*. 8
TIR *Thermal infra-red*. 8
UVE *uninformative variable elimination*. 37
VIS *visible light*. 7
YOLO *You-only-look-once*. 136

Chapter 1

Introduction

Hyperspectral imaging (HSI) systems provide rapid, accurate and economical solutions for many tasks in the food industry. Thus, HSI for food has become an active area of research in the modern food industry. Given the advancement in computer vision technologies, such as deep learning models, HSI has promising potential for this industry. In this thesis, the robustness of this kind of imaging system is investigated and evaluated for meat products.

This chapter overviews research conducted during the PhD journey. First, Section 1.1 provides the background of existing methods for the quality and safety evaluation of food products. In Section 1.2, the significance of, and motivation for this research are described. Then, we define the main objectives of this research and their applications in Section 1.3. Section 1.4 demonstrates the main research contributions. Finally, Section 1.5 outlines the structure of this thesis.

1.1 Background

This section reviews three approaches that are commonly used for the evaluation of food products. In addition, this section contains a brief description of how imaging systems are applied in this field of research.

1.1.1 Lab-based technologies

Food quality and safety attributes are usually defined based on chemical (e.g., protein, fat and moisture), microbiological (e.g., freshness and spoilage), sensory (e.g., colour, tenderness, marbling and flavour) and technological attributes (e.g., pH and water-holding capacity); see [1–8]. Practically, these attributes are still assessed by lab-based measurements or by well-trained assessors as in the case of sensory attributes. In fact, a lab-based assessment provides a good evaluation of many quality grading attributes and for a wide-range of food products. However, this kind of evaluation is subject to human errors, and is destructive, time-consuming and inconsistent.

1.1.2 Spectroscopic technologies

Spectroscopic measurements have gained increasing attention in food research as a non-destructive method for raw material analysis and the safety and quality grading of food products. Spectroscopic methods measure the optical properties of single points on a sample's surface (i.e., quantifying the interaction between light and the chemical composition of a sample across a specific range of electromagnetic wavelengths); they map those properties onto quality and safety attributes. Such properties can be defined by reflectance, absorbance or the scattering of light at specific electromagnetic wavelengths [9–11].

Compared with lab-based methods, spectroscopy has advantages such as chemical-free measurements or short measurement time with limited sample preparation. They can be used for estimating more than one attribute by the same measurements. Practically, optical properties of a sample are correlated with some quality or safety attributes for building a prediction model by the measured optical properties (i.e., reflectance, absorbance or the scattering of light). Then, this model is used for predicting the pre-defined attributes of a new set of samples.

Consequently, spectroscopic technologies are successfully applied to a wide range of food products for quantifying the quality or safety of these products. For example, they are used for quality evaluation of meat products [12–14], fish [15], poultry [16] or other kinds of food [17, 18].

Spectroscopic technologies have disadvantages regarding the non-availability of spatial information, the non-inclusion of small-sized objects into the analysis, missing flexibility in measuring particular spectral information and an inability to generate distributions of attributes [19].

1.1.3 Imaging technologies

During the past 20 years, many research efforts have been directed toward imaging and sensing systems for grading the safety and quality attributes of food products. These systems assess physical measurements of a spatial point in the field-of-view of the sensor, then reconstruct these measurements into a *2-dimensional* (2D) image array or *3-dimensional* (3D) image volume. Then, these images are used for estimating and predicting a wide range of food quality attributes.

Imaging technologies are also considered as non-destructive techniques for predicting the quality attributes of food products. Practically, many technologies were successfully introduced based on different principles such as ultrasound technology [20], *Computed tomography* (CT) scanning [21], conventional digital images [2] and HSI systems [11].

Ultrasound imaging is used for measuring the physiochemical properties and the chemical composition of food products. It has been used for estimating the chemical composition of meat products such as for beef [20], lamb [22,23], or pork [24]. In addition, CT scanning has also been successfully used for evaluating lamb carcasses [21] and beef products [25] based on a sequence of cross-sections of X-ray images of a product.

This technology is not efficient in this field due to the costs caused by evaluation time and used tools [21]. Conventional computer vision (i.e., based on recorded RGB images) is used to assess image quality attributes. It deals with spatial information, an approach not addressed by single-point spectroscopy.

Conventional computer vision has limitations regarding spectral information. A colour image represents only reflectance values for three spectral distributions of the *visible light* (VIS), namely for blue, green and red. The studies described in [26–28] show plenty of applications of colour images for the assessment of food quality. In these applications, colour image spaces were used to classify or predict the quality of food products based on predefined quality parameters.

Colour images provide support in solving the spatial information problem; they have limitations in covering spectral information because some of the quality features might be located at other wavelengths such as in the *near infra-red* (NIR) spectral domain. Because colour images only represent the blue, green and red components of VIS, they are not able to solve complex identification problems for food quality concerns.

As a logical extension of both spectroscopy and colour imaging, HSI systems enable the integration of the main advantage of spectroscopy (spectral information) with the main benefit of the conventional colour image (spatial information). From this integration, the HSI system is able to produce the quality attributes from the spectral information and identify where this prediction is located in the sample.

Moreover, HSI systems facilitate the visualization of objects and their chemical distribution (chemical image and classification map). In general, the HSI system has information about external attributes (spatial information, such as shape, and spatial distribution) and internal attributes (spectral information). From these two types of information, we[1] characterize the physical and chemical features of the objects in the image. So, HSI systems are more reliable than conventional imaging systems and spectroscopy technologies.

Practically, HSI systems are able to merge existing computer-vision technologies (e.g., considering RGB images as input) with chemometrics analysis for solving various tasks such as food safety, quality grading or classification. Thus, HSI systems

[1]The use of "we" throughout this thesis is purposeful. It is used to involve the reader with the thesis as recommended by Knuth et al. [29].

have provided robust, rapid and non-destructive solutions in many research areas, for example in agricultural and remote sensing [30], medical imaging [31] and food processing [19, 32].

In the implementation of an HSI imaging system, the spectral range, within the electromagnetic spectrum, is very important and closely associated to targeted applications of the system. There are several ranges that can be considered such as [43]: (1) VIS range (400 - 700 *nm*). (2) NIR range (700 - 1400 nm). (3) *Short-wave infra-red* (SWIR) (1400 - 3000 *nm*). (3) *Infra-red* (IR) range (occupying the largest range in infrared spectrum and approximately is defined in range of 3000 - 30000 *nm*).

The electromagnetic radiation will interact in very specific ways with a sample depending on these spectral ranges and provide information related to several attributes of materials such as chemical composition, physical properties or sensory attributes. Thus, the selection of one of these ranges is application-dependent, that is, it depends on the target material and the attributes that need to be measured.

The NIR spectral range has shown to be a rich source of information about chemical and structural information of meat [11]. It has been used for measuring many attributes related to meat quality and safety in the meat industry [60–63, 65–68, 94]. The NIR spectrum can provide characteristic information of organic molecules of meat, which involves the response of changes in vibration modes molecular bonds of CH, NH, CO, and OH overtones [41]. For example, the effect of moisture can be observed at 970 and 1450 *nm*, while the fat was observed around 1210 and 1900 *nm* [41].

The SWIR spectra is associate with overtone of molecular vibrations, with fundamental modes in the infrared. It can provide significant information related to organic materials. SWIR HSI imaging has been used in agri-food and remote sensing applications such as predicting moisture, oil, oleic acid contents, and general quality assessment of grains [42]. Similarly, *Thermal infra-red* (TIR) imaging shows a promising success in wide range of application including industrial food processing applications [43]. The principle of TIR is based on use the infrared radiation emerging from the material of interest [43]. The TIR can be based on detection of radiation in short wave to long-wave infrared [43]. The long-wave infrared systems exhibit maximum sensitivity around room temperature, while the peak sensitivity in mid-wave infrared systems is observed at much higher temperatures (e.g., 400 °C). Differently of HSI, The TIR system is based on conversion of the radiation emitted in give spectral range (e.g. 8–12 μm) into an image [43]. The resulting image shows the temperature profile of the sample.

Snapshot implementation of these imaging systems (i.e., SWIR and TIR) is very promising field of research. These snapshot implementations could have advantages of: (1) Enhance the acquisition time of the image to achieve high frame rates.

(2) Commercialize these imaging systems for particular applications depending on where the significant bands are located in these ranges, which can be positively reflected on the price of these imaging systems. However, these implementations need lots of research efforts regarding the physical calibration of sensors that maximize a significant response at each particular band. Moreover, the current state of these imaging systems makes the mass production of low-cost, consumer-grade cameras cost-prohibitive, and the first-wave of mass-produced snapshot spectral imaging cameras could be made with silicon CCD and CMOS focal planes such as those used in digital cameras[53].

In this research, we use three HSI imaging systems including three ranges in the electromagnetic spectrum: VIS, NIR, and Visible-NIR ($548 - 1,701\ nm$) range. In fact, the reason behind selecting these ranges can be summarised as follow: (1) Form the literature, these ranges provide a significant spectral information regarding the properties of red-meat types. (2) The availability of imaging systems that cover these ranges in the market with reasonable prices. (3) These imaging systems are already available in the research facilities that are provided for this research.

1.1.4 Real-time requirements for HSI system in food processing

Video rate is defined as the number of frames (images) that the camera can collect in one second for meeting the requirements of the human vision system, it is measured as *frame per second* (fps). In computer vision, typically, RGB cameras can acquire images at rate of 30 image per second or higher. The 30 fps is the most used frame rate to display sequence of frames as video clip due to human vision system. In computer vision algorithms, the definition of real-time depends on the ability of the algorithms to process number of frames in a second without buffering. Thus, the processing time of a frame (image) should be in 1/30 of a second or less to qualify as a real-time system, where the timing constraint in this case is to process 30 frames in a second for displaying the output as video clip.

The definition of 'real time', in the context of food processing applications is slightly different from computer vision applications. It is defined as time required to scan a product in time frame that enables a decision to be made and an action realized on the scanned product (e.g. assessing whether there is a foreign object in a product before it is packed). Thus, assessing whether HSI enables a real time evaluation of food should take in consideration the following constraints: (1) The acquisition time of an HSI image, that is is usually greater than RGB acquisition time due to the amount of data in the HSI image (i.e., the hypercube) [45]. For example, line scanning HSI imaging system collects image in a time frame of seconds not in milliseconds like in RGB digital imaging [45]. (2) Specifications of the production

line such as the speed of the processing chain and any manual inspection processes prior to scanning the products [45,46]. Usually, production lines in the food industry can process quite few products in a second or minute [45, 46]. (3) The computation time of the data analysis and visualization [45, 46].

In conclusion, the real-time requirement for HSI system in food processing are application dependant and it is controlled by many factors related to the specific industry requirement. In this thesis, the time involved in the processing of an image towards an information (e.g. presence/absence of a foreign object) is reported and time taken for acquisition of HSI is described in the corresponding material and methods description. These are discussed accordingly in regards to constraints they impose towards real time applications.

1.2 Motivations

Meat is an essential component of food for humans worldwide. It strongly affects the human dietary system due to its high nutritional value. Thus, research efforts are directed toward safety and novel preservation technologies, techniques for grading its quality and safety, and automated monitoring systems for its quality and safety. This motivates us in this research to focus on meat products as a specific area in the general food technology research area.

Food fraud is a growing concern from both the industry and the customer's point of view. In fact, food fraud has become such big business in the world that companies are losing money and, at the same time, consumers are losing faith and being put at risk. It has been estimated that food fraud costs the global food industry around US$30-40 billion every year [33]. Recently , food fraud incidents have been increasing across the world; thus, more efforts are needed to provide efficient, rapid, non-destructive technologies that can protect both industry and customers. *Economically motivated adulteration* is a specific type of food fraud. It involves adding a material in a product to increase the apparent value of the product for economic gain. In financial terms, this kind of adulteration costs the global industry around US$5-10 billion every year [34].

Fake lamb products recently appeared as an example of economically motivated adulteration. News media throughout the world reported on this phenomenon and its effects on consumers, industry and the brand provider's reputation [35–38]. In this case, fake lamb products were being sold as New Zealand lamb products in China markets [38]. Developing rapid and non-destructive systems for automating the detection processes for this kind of adulteration can protect both customers and the industry.

Food inspection processes aim to ensure that food products are safe, wholesome and correctly labelled and packaged. Regarding meat, there are two major types of inspection processes: product-based and carcass-based inspections. In the case of meat product inspection, the prevention of recalls is an important aspect [39]. Foreign matter (i.e., foreign objects) is physical contamination; glass, metal or plastic objects can accidentally fall into meat products during processing. Statistics published in [39] reveal that foreign matter in food products is the third most common reason for food recalls. This matter is distributed as follows: 33% metallic objects, 29% plastic objects, 24% glass objects and 14% other materials. Meat is considered one of the food types that have a high probability of recall occurrences, representing 25% of food recalls. The cost of recall incidents is high in the meat industry [39, 40]. While technologies exist for metal detection, other types of material remain challenging.

In light of this, combining the technologies of both computer vision and HSI produces efficient, intelligent, automated systems for detecting these foreign objects in meat products. Such systems can fully use the spectral information for classifying the image content based on the material type. At the same time, computer vision can provide an accurate localisation of these objects.

As a technical motivation, new hyperspectral imaging sensors, called snapshot hyperspectral imaging sensors, have been successfully designed and recently introduced into the market as the latest trend of HSI technologies. These new sensors have two novelties: (1) Hyperspectral video allows the collection of a whole hyperspectral data cube at a rate of up to 170 cube/sec (2) Mobile HSI systems do not require adjusting or moving the samples for imaging, which means that these sensors can be completely portable devices for both indoor and outdoor applications.

In fact, snapshot HSI systems record reduced spectral information compared to standard line-scanning HSI systems but at higher speed (video rate) and in the form of a portable system (mobile HSI system). These new sensors open the door to a range of applications for HSI, especially in the food industry that requires real-time processing and portable imaging systems.

Due to the above, these particular problems and their applications (i.e., red-meat authenticity and foreign matter detection in meat products) motivated this research. Moreover, the adaptation of snapshot HSI for solving these particular problems has motivated this research from a technical and theoretical point of view. Novel ways need to be developed for dealing with the data produced by snapshot HSI. This involves several challenges for the computer vision, data management and data mining and modelling components.

1.3 Objectives

The primary objective of this research is to develop methods for addressing the interaction of textural elements (i.e., pixels that share a similar spatial appearance) and the chemical distributions (i.e., the spectral representation of the chemical composition) which are provided by HSI systems. In terms of assessment, meat authenticity and foreign matter are exemplars of important problems that involve this interaction. In this research, we use these exemplars as case studies for testing and evaluating the proposed models. The number of studies addressing this interaction is limited. Thus, addressing this problem, through identifying several applications across the meat industry as well as other food industries, is one of the motivations of this thesis. The specific objectives of this research are as follows:

- To address spatial variation and spectral information and a combination of the both as useful features for classifying hyperspectral data.
- To propose and evaluate methodologies for dealing with snapshot hyperspectral data including techniques for image acquisition, sampling, feature extraction and models for classifying the reduced spectral of snapshot hyperspectral data.
- To develop and evaluate models for real-time potential recall detection and localization in meat products by using the spectral and spatial features of snapshot hyperspectral images.

1.4 Contributions

In this research, we provide several original contributions to this research area by addressing our objectives and, thus, paving the way for further improvements. The following summarises the main contributions:

First, we propose a deep learning model for classifying the line-scanning hyperspectral data by combining both 1D convolutional (1D–CNN) and 3D convolutional (3D–CNN) networks. From this combination, the model is able to map spectral features (by 1D–CNN) and spatial features (by 3D–CNN) into intelligently learned features. Moreover, the proposed model is able to handle the raw inputs of line-scanning hyperspectral images without a need for any kind of preprocessing. Also, we propose a methodology for wavelength selection for spatial feature extraction. In addition, we provide a comprehensive analysis of different kinds of spectral and spatial features as handcrafted features and compare them with the learned features of the CNN model.

Second, snapshot hyperspectral data is challenging data for food processing applications, where the spectral information is much less compared to the spectral information of line-scanning hyperspectral data. In this research, we provide a technical framework for dealing with and processing this challenging data (i.e., snapshot hyperspectral data), including image capturing and correction (reflectance and illumination corrections), training points or sample selection (by combining superpixel segmentation and ground-truth images). Also, we propose a novel 3D–CNN model for feature extraction and classifying the snapshot hyperspectral data. The proposed classification framework of snapshot hyperspectral data was evaluated by quantitative analysis with state-of-the-art models and also by visual representation for more investigation and confidence.

Third, we investigated the problem of identifying and localizing physical contaminants in red-meat products, especially in the case of foreign objects on the meat surface. To address this case, we propose a novel cascaded deep learning framework for object detection and localisation. The proposed model was analysed and evaluated in two steps: an evaluation step with images and objects used for monitoring the training processes; and a testing step with a totally new set of images of a new set of meat samples and a new set of objects. Based on the testing results, the proposed model showed robustness and novelty compared to the state-of-the-art models.

Fourth, as far as we know, there is no source reference for a public dataset of red-meat hyperspectral images. So, we built a new dataset of images from different hyperspectral sensors; these sensors are: VIS snapshot hyperspectral sensor (covering 16 wavelengths from the visible light region), NIR snapshot hyperspectral sensor (covering 25 wavelengths from the near-infrared region) and line-scanning hyperspectral sensor (covering 230 wavelengths from both visible near-infrared regions). We collected around 100 GB of hyperspectral data of three types of meat (lamb, beef, pork) and of the same meat sample with simulated foreign objects. We will continue to work on this reference dataset for publication in the near future. Based on this reference dataset, we provide benchmark results as a comprehensive comparison between these three hyperspectral sensors. This benchmark shows, in a quantitative form, the robustness of each sensor on the same research problem (i.e., red-meat classification problem).

1.5 Thesis outline

The remainder of this thesis has six chapters. It is organized as follows:

- **Chapter 2:** Presents a general background about hyperspectral imaging ba-

sics, including image structure, acquisition and sensing methods, reflectance calibration and the experimental setup of the developed HSI systems for this research.

- **Chapter 3:** Reviews state-of-the-art methods of hyperspectral imaging for food processing and other applications.
- **Chapter 4:** Presents novel models for detecting the adulteration in red-meat products by line-scanning hyperspectral images. In addition, it provides a comprehensive analysis and comparison between spectral, spatial and spectral-spatial features.
- **Chapter 5:** Presents novel models for classifying the snapshot hyperspectral data of red-meat products as a case study. In addition, this chapter provides benchmark results as comprehensive comparisons between three types of hyperspectral imaging systems: NIR snapshot hyperspectral data; VIS snapshot hyperspectral data; and full range (i.e., VIS-NIR) line-scanning hyperspectral data.
- **Chapter 6:** Presents novel deep learning models for novel objects detection as a simulated case for physical contaminant detection in red-meat products.
- **Chapter 7:** Concludes this thesis by providing a summary of the contributions of this research. In addition, it provides suggestions and directions for future research in this field.

Chapter 2

Hyperspectral Imaging

The main goal of HSI imaging is to obtain a spectrum for each pixel in the image of a scene, with the purpose of finding objects, identifying materials, detecting processes or predicting the chemical properties of materials. The pixel values in an HSI image are interpreted as a representation of the chemical composition of the materials in the scene. This representation is obtained by sensing interactions between the material surface (or through the material) and light beams.

This chapter demonstrates the basic knowledge about this kind of imaging technology; it is organized as follows. Section 2.1 shows the structure of hyperspectral images. Hyperspectral image acquisition and generation methods are described in Section 2.2. Then, Section 2.3 presents the sensing approaches of both line-scanning and snapshot images. Finally, Section 2.4 presents the tools, cameras, equipment and experiment setup that are used in this research.

2.1 Hyperspectral image structure

HSI systems are valuable tools for visualizing the chemical components of materials by means of an image, providing detailed information about their types and shapes. In general, an HSI image is either a vector-valued 2D image or a stack of 2D images (the *layers* of the stack).

Assume that we record for discrete wavelengths λ_1 to λ_n and at coordinates $(x, y) \in \Omega$, where Ω is a 2D rectangular grid. In this case, an HSI image H is defined as a 3D image *hypercube*

$$H : D_x \times D_y \times D_\lambda \to \mathbb{R} \tag{2.1}$$

where $D_x \times D_y$ represents the space of the spatial coordinates (x, y) of each layer, and D_λ is the space of the selected discrete wavelengths across a specific range of wavelengths. Thus, H is a stack of 2D images (or layers) at specific wavelengths, going from λ_1 to λ_n.

Figure 2.1 shows a schematic representation of a hypercube, as a stack of 2D images at specific wavelengths ranging from λ_1 to λ_n. As shown in Figure 2.1, left, each layer is also called a *band* and defined as a mapping $H_\lambda : D_x \times D_y \to \mathbb{R}$.

Figure 2.1, right, illustrates the pixel vector in a hyperspectral data cube. The pixel is a 1D signal of the light radiance intensities. These intensities represent the interaction between the light beams from the source light and the material composition. So, the pixel values are considered as a spectral signature of this material and its unique value for each type of material.

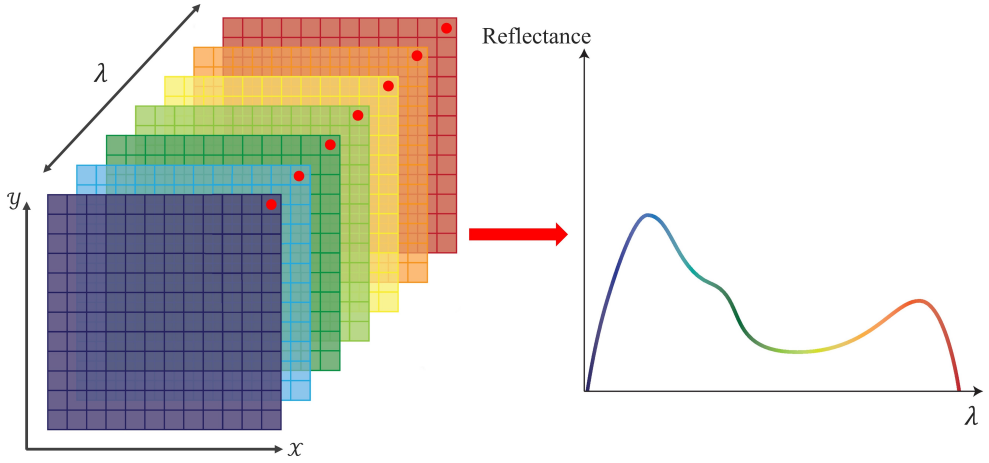


Figure 2.1: Schematic representation of a hypercube; red points in this hypercube show one pixel location; its spectral signature is shown on the right.

2.2 Hyperspectral image acquisition and generation

In HSI systems, the way the HSI image is acquired is very important for defining the resolution and type of information in an HSI image. There are four approaches for collecting and reconstructing the HSI image, $H(x, y, \lambda)$: point scanning (whiskbroom); line scanning (push broom), area scanning (band sequential) and snapshot scanning [19, 45, 47]. In the case of point scanning, the camera detector has a grid of $1 \times 1 \times \lambda$ cells, which means that the camera is able to collect the spectral information of a single point with a predefined spatial resolution. In other words, the camera or the sample needs to be moved across x and y directions sequentially.

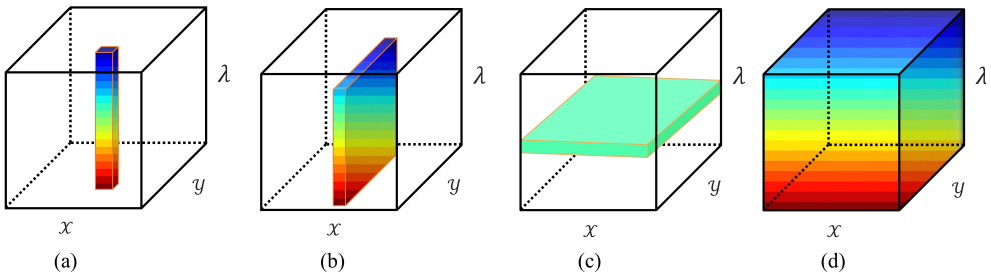


Figure 2.2: A representation of HSI acquisition approaches: (a) point scanning; (b) line scanning; (c) area scanning; and (d) snapshot scanning.

Figure 2.2 (a) demonstrates the point scanning method. The main advantage of point scanning is the ability to obtain a very high spectral resolution (up to thousands of wavelengths), so it is mostly used for advanced problems in food quality grading and medical applications. However, it is very slow; mostly this is because point scanning needs a lot of time (hours) to collect one HSI image. In the case of line scanning, one spatial dimension is fixed, the y dimension as illustrated in Figure 2.2 (b), and the other dimension is free for scanning by moving the camera or sample. Thus, the camera detector, in this case, has $1 \times y \times \lambda$ cells. Line-scanning HSI is the most used method for quality and safety inspection processes due to its suitability for conveyor belt systems. However, it has disadvantages such as needing a critical setup for adjusting the source light and the speed of the sample or the moving camera. Thus, the total time for collecting one HSI image is still greater than video rate time; taking seconds for collecting one image.

The abovementioned two methods are called spatial scanning methods, while there are another two methods called area and snapshot scanning. Area scanning is a spectral scanning method, where the spatial dimensions (x, y) are fixed and the camera acquires the spectra of a single wavelength at a time, and this process is repeated until the whole spectral range is covered; Figure 2.2 (c) shows a representation of this scanning method.

An extension of area scanning, called snapshot HSI, has been recently developed and introduced. In this case, both spatial and spectral information are acquired in one shot, as illustrated in Figure 2.2 (d). A snapshot hyperspectral system is a unique sensor for HSI where the spectral sensing unit monolithically integrates on top of a standard complementary metal-oxide semiconductor (CMOS) sensor at the wafer-level [49, 50].

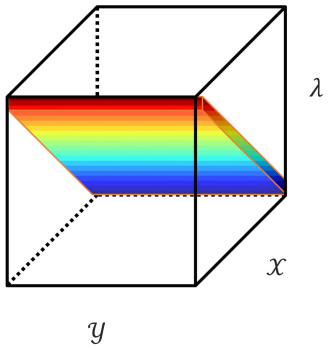


Figure 2.3: A demonstration of HSI spatio-spectral acquisition approach.

Snapshot HSI implementation helps in designing and manufacturing a product based on highly advanced HSI devices. Many applications could be used for this implementation; the different types of sensing filters provide a flexible usability in many fields. The snapshot HSI systems can measure and identify the chemical spectral signatures for the materials. Thus, snapshot HSI systems are useful for many applications such as object tracking based on spectral properties, object detection and recognition in real-time mode, finding foreign objects, material sorting, food inspection and those in the field of medicine.

Recently, the spatio-spectral scanning method was proposed as a combination of strengths of line scanning and area scanning [54]. In spatio-spectral scanning, the camera is moved transversely to the slit of a basic spectroscope [54]. Thus, spatio-spectral scanning compensates for the corresponding weaknesses of both line scanning and area scanning, where static and mobile acquisition modes are possible in spatio-spectral scanning [54]. Figure 2.3 demonstrates the spatio-spectral scanning approach.

2.3 Sensing of hyperspectral data

Spectral information obtained by HSI systems depends on how the light interacts with the sample surface or its internal structure. There are three common modes for sensing the spectral information in HSI called reflectance, transmittance or interactance; as demonstrated in Figure 2.4. The light position and the angle (with respect to the camera) defines these sensing modes. In reflectance mode [19, 45, 47], the sample is illuminated by a special type of light, then the light beam penetrates the sample and interacts with the internal composition of the sample, then light is

reflected as energy irradiance values towards the camera detector; Figure 2.4 (a) illustrates this type of sensing. The signals reaching the camera represent the chemical composition of the sample and its structure. HSI systems, in reflectance mode, are the most used systems for many HSI applications, especially for meat and food processing [19, 47, 48].

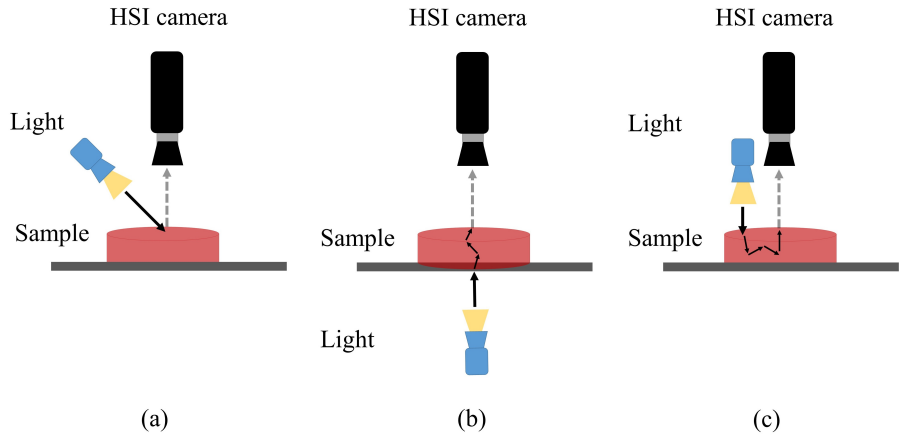


Figure 2.4: A representation of HSI sensing approaches: (a) reflectance mode, (b) transmittance mode, and (c) interactance mode.

In the transmittance mode [19, 45, 47], the light source is set in the opposite direction to the camera as shown in Figure 2.4 (b). In this case, the camera detects the amount of light that penetrates the sample (usually they are weak signals); these signals reflect valuable information about the internal structure and composition of the sample. The third mode, interactance [19, 45, 47], is a combination of reflectance and transmittance modes. As shown in Figure 2.4 (c), the camera and light source are directed into the point on the sample.

2.4 Experiment setup of the HSI systems

This section presents the components and the experimental setup of two HSI systems which were used for collecting our datasets: the line-scanning HSI system and the snapshot HSI system.

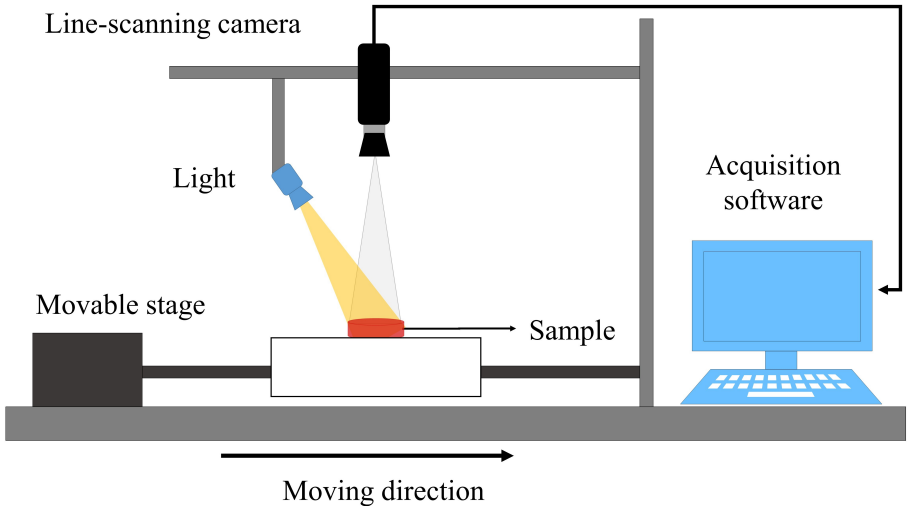


Figure 2.5: Schematic demonstration of the main components of the line-scanning HSI system.

2.4.1 Line scanning hyperspectral imaging system

A line-scanning (push-broom) HSI system, in the reflectance sensing mode, was developed and then used for image acquisition. The line-scanning system, as shown in Figure 2.5, consists of a moving table, a hyperspectral camera with a Headwall spectrograph (Model 1003B-10151, Headwall Photonics, Fitchburg, MA, USA) that captures 320 spectra in a line. Each spectrum has 235 wavelengths ($N_{bands} = 235$) in the spectral range of $548 - 1,701$ *nm* (this range covers the visible light and NIR regions in the electromagnetic spectrum) with spectral resolution of 5 *nm*; spectral resolution is approximately the difference between any two contiguous wavelengths in nano-meters within the whole considered spectral range. A 25 mm lens with an aperture (f/stop) of 2.8 was used to ensure that variation in uneven surface of the sample were still under the field of view of the camera.

The illumination system has one halogen lamp light source (JCR 21V 150W/AL Japan 2DB) and is designed to distribute the light uniformly over the line detected by the camera. This system allows for light power delivered to the samples to be adjusted. In this case, the power is adjusted using a white reference tile where the highest intensity detected in the white reference tile is set as 85% to avoid saturation of the detector. This is done to prevent that regions on the sample could saturate the detector.

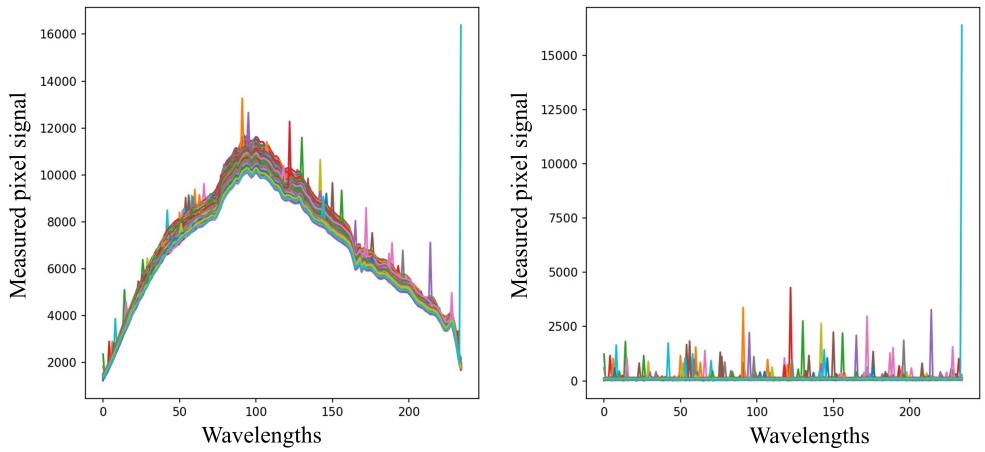


Figure 2.6: Distribution of intensities of white and dark references of line-scanning HSI. *Left*: the distribution of the reflected light intensities of the standard white board at each pixel in the line (320 pixels) at each wavelength. *Right*: the distribution of the intensities detected with the camera covered off (dark situation).

Empirically, the following parameters were adjusted to improve the quality of the images: (1) The distance between the camera and the target sample was set to 36 *cm*. (2) The exposure time was set to 34.7 *ms*, and the translation speed of moving table was adjusted to 11.1 *mm/s* to ensure that pixels were square and with spatial resolution of about 0.4×0.4 *mm* for one pixel. (3) The quality of light, and the position and intensity of the light source are very important factors.

The distance between camera and light was set to 32 *cm*, which obtains light beams directed to the sample at an angle of 45 degrees to the camera detector. Both the position and intensity of the light are adjusted by monitoring the standard deviation of the light profile as shown in Figure 2.6, left; a low standard deviation means a more stable light fairly distributed on all pixels in the spatial line.

In line-scanning HSI, the camera scans a line of 320 (N_{col}) pixels and for each pixel a spectrum is acquired. To obtain a hyperspectral image in this system, the sample is moved with linear speed in a direction orthogonal to the direction of the scanned line. In this way, the whole sample is scanned and then reconstructed as 3D image R with a shape of ($N_{row}, N_{col}, N_{bands}$). Then, all images were calibrated for obtaining the reflectance values as follows:

$$H_\lambda(x,y) = \frac{R_\lambda(x,y) - D_\lambda(x,y)}{W_\lambda(x,y) - D_\lambda(x,y)} \tag{2.2}$$

The calibrated image reflectance H is obtained from the raw image irradiance R by

using a dark reference image D and a white reference image W. Parameters x, y and λ are 3D image coordinates of column, row and wavelength, respectively, with $x \in D_x$, $y \in D_y$, and $\lambda \in D_\lambda$; 3D images H, R, D and W are defined by 2D layers (e.g., R_λ), for all the contributing λs.

The line-scanning array detector has 320×235 ($N_{col} \times N_{bands}$) cells defining the field of view of camera as one spatial line. Thus, the reference data (i.e., white and dark references) of the detector was collected for each cell as one spatial line of N_{col} pixels with their spectra. The resulting white and dark data have a shape of $1, N_{col}, N_{bands}$. To match the shape of these data with the shape of R image, W and D are defined by repeating each array N_{row} times to obtain 3D images with a shape of ($N_{row}, N_{col}, N_{bands}$). This process means that the white and dark calibration of the raw HSI image is computed line by line. After this calibration process, the first and the last five layers of H were removed due to a low signal-to-noise ratio, defined by the mean divided by the standard deviation.

2.4.2 Snapshot hyperspectral imaging system

In this research, we used two snapshot HSI sensors for developing a snapshot HSI system: MQ022HG-IMSM5X5-NIR (NIR) and MQ022HG-IM-SM4X4-VIS (VIS)[1].

The first sensor (i.e., NIR) is a mosaic image with a per-pixel design. The mosaic image is a rectangle of $2,045 \times 1,080$ for width and height, respectively, which represents the raw image data; Figure 2.7 (a) shows an example of the raw mosaic image of a beef sample. The mosaic image is structured by repeated sub-grids of 5×5 pixels (called micro-pixels). Figure 2.7 (d) shows an example of one of these micro-pixels and its wavelength values. Each micro-pixel represents the signals of 25 spectral pattern filters in the range of $672.74 \sim 957.49$ *nm*. Then, the raw mosaic image is reconstructed by re-folding each micro-pixel into a 1D vector as a spectrum, for obtaining a hypercube of $409 \times 216 \times 25$ for width, height and wavelength, respectively as shown in Figure 2.7 (b).

The second sensor (i.e., VIS) is structured in the same way as the NIR sensor, where the raw mosaic image size is $2,048 \times 1,024$ for width and height, respectively, the micro-pixel size is 4×4 of 16 spectral pattern filters in the range of $466.9 \sim 639.3$ *nm*, and the reconstructed hypercube has a shape of $512 \times 256 \times 16$ for width, height and wavelength, respectively.

Then, we developed an HSI system by using these snapshot HSI sensors (i.e., NIR and VIS). Figure 2.8 shows a schematic representation of the main components of this HSI system. The snapshot HSI system consists of two snapshot HSI cameras. Other components include an illumination unit of six tungsten-halogen lamps

[1]Both manufactured by Ximea with an image on a chip from IMEC [49, 50]

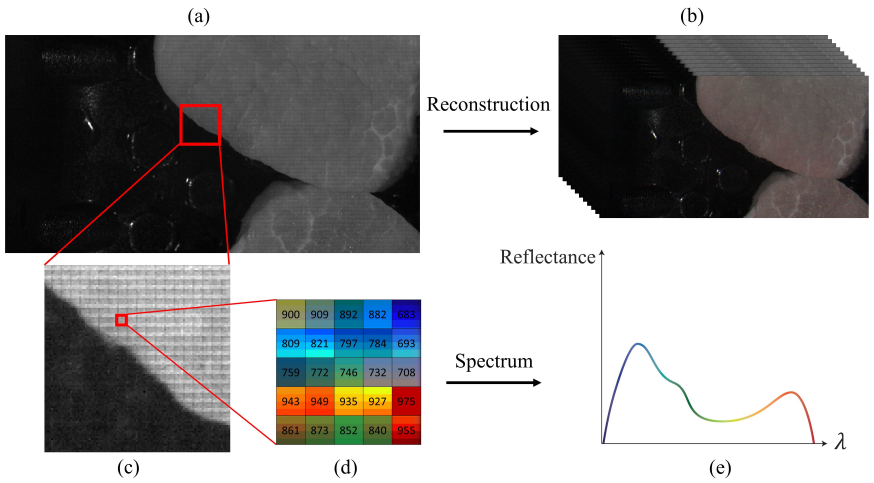


Figure 2.7: Representation of snapshot HSI image structure: (a) the raw mosaic image; (b) The reconstructed HSI hyperspectral cube; (c) and (d) the micro-pixel and the spectral patterns; and (e) a spectrum of a micro-pixel.

(each of 20 watts), a conveyor belt and image acquisition software (HSImager). The distance between the camera and the conveyor belt was set to 35.5 *cm* to enable the wider sample to be within the field of view of the camera. The exposure time of the snapshot cameras was adjusted to 2 *ms* and 3.9 *ms* for NIR and VIS, respectively. These values were adjusted based on reflectance of the white reference tile in order to avoid saturation of the detectors.

The speed of the conveyor belt was adjusted empirically to be 5.5 *cm/s* to prevent blurring of the images. This resulted in a spatial resolution of 0.27×0.27 *mm/pixel* for each pixel in the resulting 3D HSI images. It should be noted that the samples could have been scanned at a stationary position; however, a conveyor belt was used to simulate a situation where samples can be evaluated automatically, for example in a meat processing plant. The resulting VIS snapshot image covers 16 wavelengths ($N_{bands} = 16$) in a range of $467 - 639$ *nm* with spectral resolution of ≈ 10 *nm*, while NIR snapshot image covers 25 wavelengths ($N_{bands} = 25$) in the range of $673 - 957$ *nm* with a spectral resolution of ≈ 10 *nm*.

In snapshot HSI system, the camera and light sources (i.e., the six tungsten-halogen lamps) were adjusted by monitoring the light distribution on the white reference tile in order to avoid saturation of the detectors. The positioning of the halogen lamps was empirically arranged as three lamps on the left of the cameras and three lamps on the right of the cameras; Figure 2.9 demonstrates this arrange-

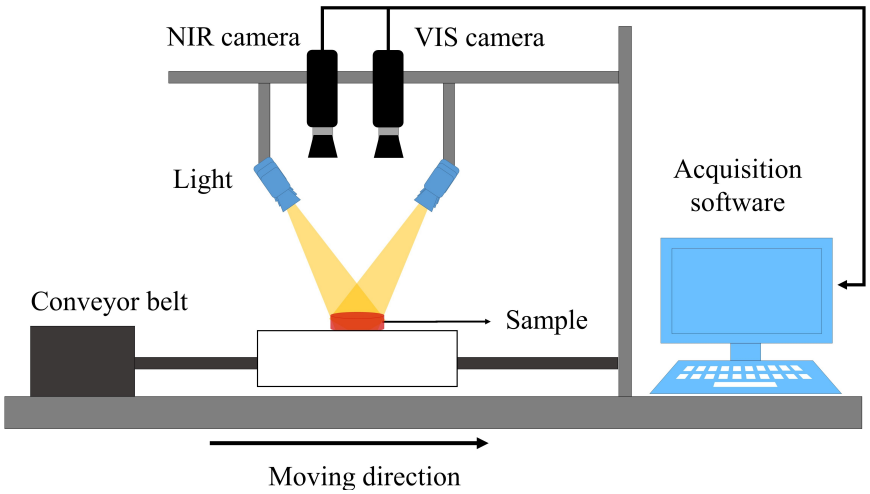


Figure 2.8: Schematic demonstration of the main components of the used snapshot HSI system.

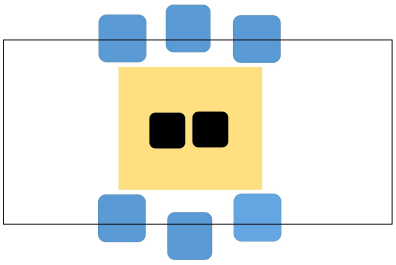


Figure 2.9: Schematic demonstration of the top view of snapshot HSI system shows the arrangement of the light units in the snapshot HSI system. The colours Blue, Black and Yellow represent the light units, cameras and field-of-view, respectively.

ment. Then, the light direction of each lamp was adjusted to obtain a well-light distribution (i.e., on the white reference tile) covering all the field of view. In our setup, the well-light distribution was defined by using the following parameters: (1) Adjusting the exposure times of cameras to avoid any saturated pixels. (2) Adjusting the lamps' directions and angles to avoid high dark regions in the field of view (low standard divination in spatial domain). Figure 2.10 shows the distribution of the detected intensities of dark and white references of NIR and VIS HSI sensors after adjusting the imaging system.

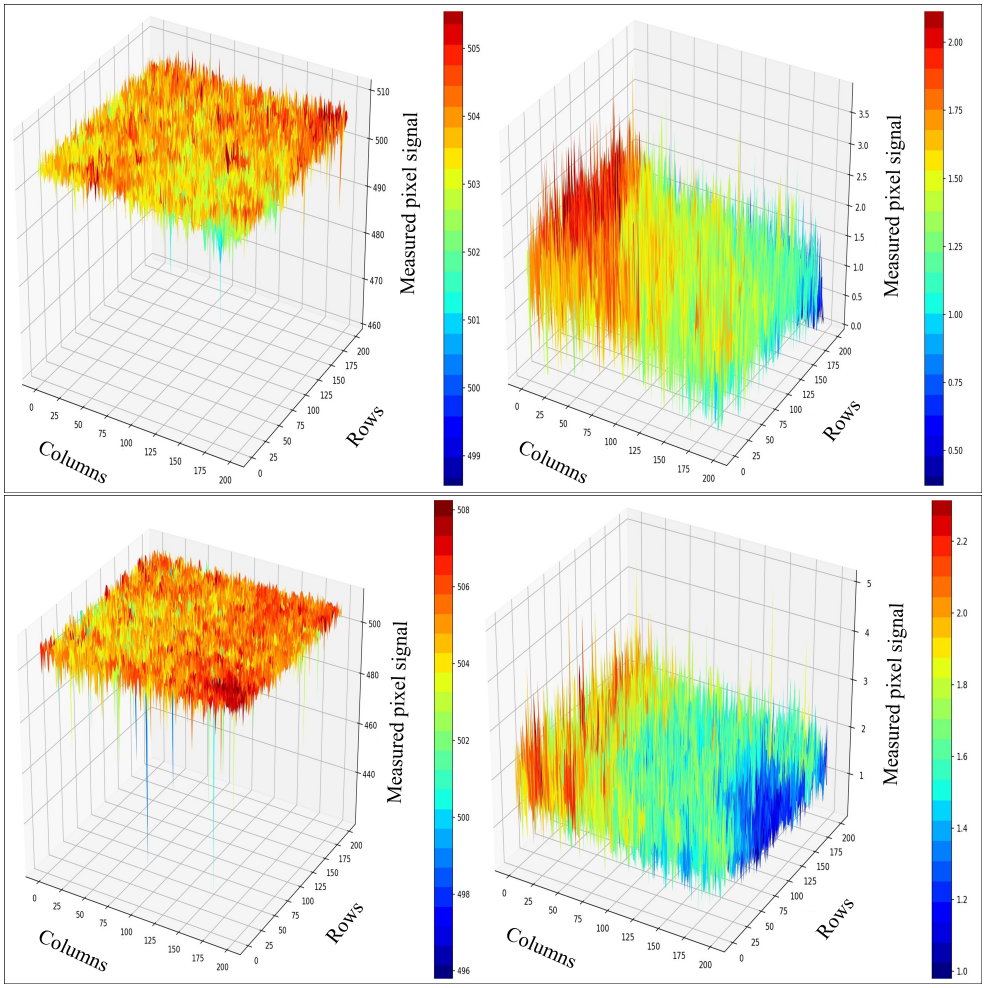


Figure 2.10: Distribution of intensities of white and dark references of snapshot HSI. *Top*: the distribution of the reflected light intensities (white) and the detected signals when the camera was covered off (dark) of the NIR sensor at wavelength of 860 *nm*. *Down*: the same distributions of VIS sensor at a wavelength of 589 *nm*.

In fact, the wavelength locations in micro-pixels (i.e., in the raw mosaic image) are spatially separated and distributed. This distribution is fixed from the manufacturer where each location is a filter maximizing the response at a particular wavelength; Figure 2.7 (d) shows the wavelength locations in the micro-pixel of NIR snapshot sensor. Thus, the micro-pixel represents the spectra of a single spa-

tial location and the wavelength locations are spatially separated in the sensor level (in micro-meter), which means that they do not have a significant impact on the reflectance calibration of the resulting hypercubes.

Therefore, the reflectance calibration process is applied on the hypercubes (HSI images) after converting the raw 2D mosaic images into 3D images. Thus, we assume that the data of all wavelengths were collected for the same spatial location (one pixel), which means that the collected signals represent the spectra of that spatial location.

The resulting snapshot HSI images (in irradiance) were preprocessed for obtaining reflectance values considering a correction with respect to the illumination distribution by using prospective illumination correction [52]. The prospective illumination correction uses two reference images acquired for defining the white and dark references of cameras [52]. It should be noted that both white and dark reference data are 3D array (HSI images) representing the reference data for each cell in sensor array, while the snapshot camera is able to collect all spectral information of the sample without moving the sample (i.e., at stationary position). The white reference image is used to define maximum possible response of a spectrally and spatially uniform white surface, and also for defining the level of saturation to the sensor (85%). The dark reference image is used for removing static noisy signals resulting from the sensor.

The reflectance values in the snapshot images were computed in a way similar to those in line scanning, but the reference data of snapshot sensors have the same size and shape as raw images (i.e., pixel by pixel correction). Thus, the reflectance correction of snapshot images is computed as follows:

$$H_\lambda(x,y) = \frac{R_\lambda(x,y) - D_\lambda(x,y)}{W_\lambda(x,y) - D_\lambda(x,y)} \tag{2.3}$$

The image reflectance H is obtained from the raw image irradiance R by using a dark reference image D and a white reference image W; the resulting reflectance images are in the range of $[0.0, 1.0]$

The saturation level (or pixels) is defined by a threshold ($\tau = 511$) in the scale of the white reference image. Saturated pixels are the pixels that have the maximum sensor response; these pixels do not have any useful spectra. To label and remove any saturated pixel by the threshold τ, the resulting reflectance images need to be transformed into the scale of raw data (0 to 600). Thus, a normalization constant is added to Eq. (2.3) to recover the resulting images into the original scale of raw images (i.e., same scale as in white reference image). Thus, reflectance calculation in

Eq. (2.3) is further processed as follows:

$$\overline{H}_\lambda(x,y) = H_\lambda(x,y) \cdot \frac{\text{mean}\,(R_\lambda)}{\text{mean}\,(H_\lambda)} \tag{2.4}$$

where $\overline{H}$ is HSI image in reflectance with the same scale as in the raw image. Then, all pixels with values above τ were considered as saturated pixels and ignored for any further analysis and calculation. Next, the reflectance-corrected images were normalized to obtain reflectance values in the interval $[0.0, 1.0]$ as follows:

$$H^\star_\lambda(x,y) = \frac{\overline{H}_\lambda(x,y)}{\max(W_\lambda)} \tag{2.5}$$

i.e., the maximum is taken over all values at each wavelength (λ) of W. All further processing of snapshot HSI data uses those normalized reflectance-corrected calibrated images $H^\star$ as input. This normalization step is used to transform all pixels in each wavelength into same scale (i.e., $[0.0, 1.0]$), which is useful while analysing the HSI images.

Note that for snapshot images, we correct illumination over the whole image, while in line-scanning images, we correct illumination only over a single scanline. Moreover, the number of layers N_{bands} of the 3D image (i.e., number of covered wavelengths) are different from line-scanning and snapshot HSI; line scanning covers deep spectral information with $N_{bands} = 235$ while snapshot covers limited spectral information with $N_{bands} = 25$ and $N_{bands} = 16$ for NIR and VIS, respectively.

Chapter 3

Literature Review

Hyperspectral imaging systems are an advanced imaging technology due to their robustness in merging both spectroscopic and computer vision technologies. Hyperspectral data have rich information in terms of spectral and spatial information, which makes this high dimensional data a real challenge in computer vision tasks. Moreover, many models and methods need to be applied to this kind of data from different fields of research. This chapter reviews the state-of-the-art models and methods used for analysing, processing and classifying hyperspectral data.

The chapter is organized as follows. Section 3.1 presents an introduction to typical applications of hyperspectral image data. Section 3.2 shows the existing methods for analysing hyperspectral image data. The state-of-the-art classification models are critically reviewed in Section 3.3. Finally, Section 3.4 summarizes the literature by highlighting the research gaps.

3.1 Introduction

Spectral information (i.e., spectroscopic technology) is a reflection of the optical properties of a specific light on a material surface (or inside a material), which provides signals (i.e., spectra). These signals are interpreted as a representation of the chemical composition of that material so reproducing this spectral information in the shape of an imaging system is able to produce a non-destructive tool called a hyperspectral imaging (HSI) system, also called imaging spectroscopy or chemical imaging. An HSI system provides both spectral information (it tells us what the type of material is) and spatial information (due to the nature of imaging, it tells us the shape of the material and where it is in the scene).

In recent years, HSI technologies have received attention in many research areas: food processing and grading [27,32,48,51], agricultural products processing [19,55], remote sensing and landscape analysis [30, 56–59], and medical applications [31]. In fact, this large-scale application comes from the deep spectral information (if we

compare it with conventional RGB digital imaging) that HSI systems are able to provide.

In meat processing, HSI systems are widely used as a non-destructive tool for predicting many chemical attributes which are related to the quality and safety of meat products such as PH-value [60–62], water-holding-capacity [63], intramuscular fat [94], tenderness [65–67], or springiness [68]. Moreover, the HSI system has shown efficiency in materials classification or discrimination (e.g., classification between materials that visually look the same). For example, an HSI system is able to discriminate between red-meat types, meat muscle types, and meat freshness.

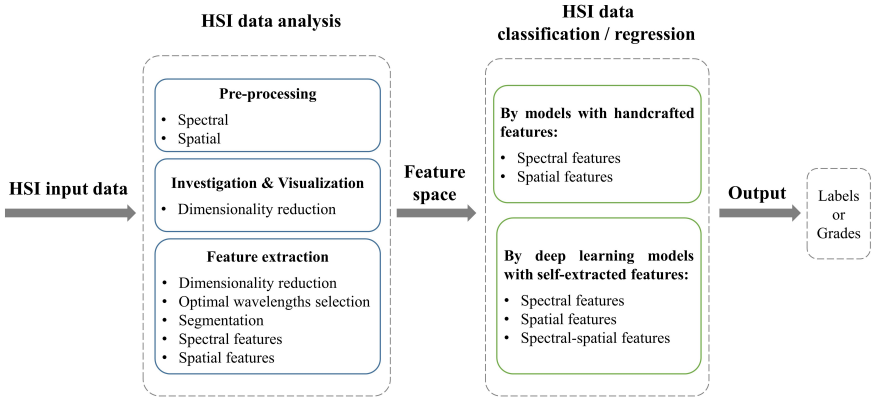


Figure 3.1: The general methodology of HSI data analysis and modelling.

In the literature, the general research methodology for processing HSI images includes the following phases: HSI data analysis and HSI data classification or regression. Each phase has its functionality and research methods. Figure 3.1 demonstrates these phases, the sub-phase of each phase and the interaction between them. In the next sections, we provide a comprehensive review of the existing research around these phases.

3.2 Hyperspectral data analysis

In HSI applications of food processing, the analysis of an HSI dataset involves preparing the dataset of images, investigating the collected data and extracting important features (i.e., spectral features, spatial features or a combination of both) for a particular classification or regression task. Thus, these steps play an important role in having an accurate classification or regression model of food processing applications.

3.2.1 Hyperspectral data preprocessing

Spectral preprocessing is used to extract a significant contribution of a specific material inside the spectra [45]. The preprocessing methods are typically used to reduce the effect of experimental variation and physical-chemical characteristics of the samples that affect the HSI system measurements [45]. There are three common preprocessing methods: spectral smoothing and derivatives [69,70], *standard normal variate* (SNV) [71,72] or *multiplicative scatter correction* (MSC) [73].

Spectral smoothing and derivatives

Smoothing the spectral data is an important step in removing the noise. The spectral signals in HSI images are defined as one-dimensional signals (i.e., the value of each pixel in the image). Thus, for reducing the effect of random noise in these signals, the simplest approach is through a moving average process (or also called spectral low pass filtering). In the moving average approach, the local means across a window size are used to compute the new values of the spectra (i.e., the smoothed spectra). Mathematically, moving average smoothing of a spectrum is computed as follows:

$$Y_j^* = \frac{\sum_{i=-m}^{m}(Y_{j+i})}{N} \tag{3.1}$$

where Y_j^* is the smoothed spectrum at wavelength j, $N = 2m+1$ is the window size (window size should be an odd number), m is half of the window size minus 1, and Y_{j+i} is the original data point at wavelength $j + i$ within the window.

The moving average approach was extended by Savitzky and Golay in 1964, thus referred to as the Savitzky-Golay (SG) method [69]. This method uses a moving window of different odd-numbered window sizes in the process, unlike the moving average approach which uses an averaging approach with the same windows. Moreover, the SG approach uses a set of predefined coefficients as a convolution window to do the smoothing calculation. Mathematically, the SG smoothing method is defined as follows:

$$Y_j^* = \frac{\sum_{i=-m}^{m}(C_i Y_{j+i})}{N} \tag{3.2}$$

where Y is the original spectrum, Y^* is the smoothed spectrum, C_i is the convolution coefficient for the i-th spectral value of the filter, $N = 2m + 1$ is the number of convolution integers (window or filter size) and m is half of the filter size minus 1.

The coefficients of the filter are computed by fitting a multi-order polynomial equation based on the least-square concept. The coefficients in zero-order polynomial fitting are the same as in the moving average method, whereas the coefficients

in multi-order polynomial least-square fitting are different and they provide shaped filter windows for data smoothing. Thus, SG smoothing is commonly used for filtering the spectrum for each pixel, where the shape of the spectrum is taken into account.

The derivatives (1st and 2nd) of the spectra are used to emphasize the spectral information. However, they increase the noise in the spectral domain [69, 70]. The SG approach can be applied to compute the derivatives of a spectrum. In this case, the derivatives are computed by using a set of coefficients that are estimated based on a least-square fitting considering a specific window size and a polynomial order [69]. The first derivative is used to remove the additive baseline, and the second derivative is used to remove the linear baseline (multiplicative), from the signals [69, 70]. The derivatives of spectra are often used for emphasizing the difference in reflectance between the spectra [74]. In addition, they are used for selecting important wavelengths that have high contributions to, or optimal bands in, a particular problem [74].

Multiplicative scatter correction

One of the challenges of an HSI system is the effect of light scattering. Light scattering could be defined as the variation of the light resulting from the sample superficial and internal structure. In HSI, it is useful to apply spectral preprocessing to the image to reduce the effects of light scattering. MSC preprocessing is used to reduce the impact of these effects from the final spectra of samples [45, 71, 73]. In fact, MSC corrects the collected spectrum (Y) (i.e., pixel values in an HSI image) with reference to an ideal or "reference" spectrum (Y_{ref}), assuming that the collected spectrum is a combination of the reference spectrum (linear combination) and some additive and multiplicative noise. This combination can be defined as follows:

$$Y = a + b * Y_{ref} \tag{3.3}$$

The constants a and b can be estimated by least-square regression. Then, the corrected spectrum (Y^*) is calculated as follows:

$$Y_i^* = \frac{(Y_i - a)}{b} \tag{3.4}$$

In the case of hyperspectral images, the reference spectrum could be obtained as the mean spectrum of the sample in the image, then the pixels individually are corrected by using Eq.(3.4).

Standard normal variate

Similar to MSC preprocessing, SNV preprocessing aims to decrease spectral variability introduced by sample morphology in HSI images [45, 71, 72]. Unlike the MSC method, SNV transformation can be applied to each spectrum individually. In the SNV method, the spectrum (i.e., pixel values in the image) is transformed into a standard version by subtracting its mean and dividing by its standard deviation [72]. Thus, all pixels share the same mean of 0 and standard deviation of 1. Practically, SNV transformation of a spectrum is computed as follows:

$$Y_i^* = \frac{(Y_i - \overline{Y})}{\sqrt{\frac{1}{n}\sum_{j=1}^{N}(Y_i - \overline{Y})^2}} \tag{3.5}$$

where Y is the original spectrum, Y^* is the transformed spectrum, $\overline{Y}$ is the mean value of the original spectrum, and N is the total number of wavelengths in the original spectrum.

As a summary of these preprocessing methods, Table 3.1 identifies the main advantages and disadvantages of each method. From an application point of view, applying one of these methods depends on the quality of the data and the target application [45].

Table 3.1: Summary of the main spectral preprocessing methods.

Preprocessing Methods		Usage and Advantages	Disadvantages and Control Parameters
Smoothing		Removes the fine noise. Manipulates the spikes. The ability for polynomial curve fitting. Applied to each spectrum individually.	Having to choose the window size and polynomial order.
Derivatives	1st	Used for emphasizing the separation between materials. Removes the additive baseline in the signal. Optimal wavelengths selection. Applied to each spectrum individually.	Having to choose the window size and polynomial order. Increases the spectral noise. Eliminates some spectral information.
	2nd	Used for emphasizing the separation between materials. Removes the linear (multiplicative) baseline in the signal. Optimal wavelengths selection. Applied to each spectrum individually.	Having to choose the window size and polynomial order. Increases the spectral noise. Eliminates some spectral information.
SNV		Reduces the effects of light scattering. Standardizes the spectral domain. Applied to each spectrum individually.	Increases the spectral noise. Eliminates some spectral information.
MSC		Reduces the effects of light scattering. Uses the mean spectral to preprocess the spectral data.	Increases the spectral noise. Eliminates some spectral information. Cannot be directly applied to each spectrum individually.

3.2.2 Dimensionality reduction and visualization

The high dimensionality of the HSI image data is a real challenge while analysing an HSI image or HSI dataset. This high dimensionality reflects negatively on the image data visualization and analysis. *Principal components analysis* (PCA) [75] is used to solve the high dimensionality problem in HSI images. Assuming that the wavelengths of an HSI image are variables, then each pixel in the image defines observations from these variables. PCA explores the relationships between the variables taking the variation of pixel values into account. Thus, PCA uses the statistic properties of hyperspectral wavelengths to examine the dependency or correlation between wavelengths. Mathematically, PCA is based on the eigenvalue decomposition analysis of the covariance matrix of the hyperspectral. First, the 3D HSI image is converted in row-wise 2D matrix ($\mathbf{X} \in \mathbb{R}^{M \times N}$) containing the pixels as M rows and the wavelengths as N columns. Thus, a pixel in the image is defined as follows:

$$\mathbf{X}_j = [x_1, x_2, x_3, ..., x_N]^\top \tag{3.6}$$

where $\mathbf{X}_j$ is a vector defining the spectra of j-th pixel in the image and N is the total number of wavelengths (or variables).

Then, the covariance matrix **Cov** of **X** is computed as follows:

$$\mathbf{Cov} = \frac{1}{M} \sum_{j=1}^{M} (\mathbf{X}_j - \bar{\mathbf{X}})(\mathbf{X}_j - \bar{\mathbf{X}})^\top \tag{3.7}$$

where $\bar{\mathbf{X}}$ is the mean vector of all image wavelengths and M is the total number of pixels in the HSI image. Next, the eigenvalue decomposition of the covariance matrix is formulated and computed as follows:

$$\mathbf{Cov} = \mathbf{ADA}^\top \tag{3.8}$$

where $\mathbf{A} \in \mathbb{R}^{N \times N}$ is a matrix showing the corresponding N dimension eigenvectors (also called base vectors or loadings vectors) of the covariance matrix **Cov** and $\mathbf{D} = diag(l_1, l_2, .., l_N)$ is a diagonal matrix containing the corresponding N eigenvalues ($\mathbf{l}$) of the covariance matrix **Cov**. The values of these matrices (i.e., **A** and **D**) are obtained by solving Eq.(3.8).

For finding the most important eigenvectors that approximate the original data, the eigenvalues and eigenvectors need to be arranged in descending order based on the eigenvalues, then the first K eigenvectors (i.e., first K rows of the matrix $\mathbf{A}^\top$) that have the significant explained variance can be selected as an approximation of the original HSI image. The approximation of the original HSI image is called the PCA image, PCA components or PCA scores of the image. In fact, the PCA score is

a linear combination between the original data and the corresponding eigenvectors and can be computed as follows:

$$\mathbf{PCA}_{scores} = \mathbf{BX} \tag{3.9}$$

where $\mathbf{PCA}_{scores} \in \mathbb{R}^{M \times K}$ is a matrix containing the first K PCA scores and $\mathbf{B} \in \mathbb{R}^{K \times N}$ is a sub-matrix from $\mathbf{A}^{\top}$ containing the first K eigenvectors as rows (i.e., the first K rows of the matrix $\mathbf{A}^{\top}$). It should be noted that the first K PCA scores are orthogonal variables and often contain the majority of spectral information in the original HSI image.

In general, PCA is commonly used in HSI data analysis of food processing applications. Practically, two concepts are used regarding PCA in HSI data analysis: (1) PCA loadings vectors (eigenvectors or base vectors) for obtaining the wavelengths that have significant impact in solving a particular classification or regression problem. (2) PCA scores for either visualizing the data or spectral feature extraction in classification and regression tasks.

In [74], a PCA model was used to classify pork meat into four quality grades based on the NIR spectral features ($900 \sim 1700$ *nm*) of an HSI system. The loadings of the PCA model were then used to generate PCA scores of the new test sample. The first three PCA scores, which have the most significant information, were merged as a pseudo-colour image. After that, the colour image was indexed into a limited number of colour ranges to generate a classification map. This approach achieves excellent results (overall accuracy = 96%) as unsupervised classification methods.

The same methodology was also applied [77] to discriminate between three types of lamb muscles: *semitendinosus* (ST), *longissimus dorsi* (LD), and *psoas major* (PM). Another usage for PCA is in initially investigating the class separation by reducing the dimensionality of the collected data into two or three dimensions [74, 77–81]. In these implementations, the scatter plots between PCA scores were used to demonstrate the classes separation. These plots provide an overview for evaluating the ability of the classification problem. As a denoising tool, in [82] a PCA model was utilized as a preprocessing tool to remove the noise from the images and to erase the missing data point inside the image. Moreover, PCA is used for selecting the *region of interest* (ROI), for example, Barbin et al., [74] implemented a method to choose the ROI using the first two PCA scores.

Although PCA has been widely used and successful in large-scale HSI applications, it has some limitations in the context of complex tasks such as multi-class segmentation. These limitations can be summarized as follows: (1) PCA is not scale invariant, the data need to be scaled and standardized, thus small-size segments could be affected by the scaling and standardisation processes. (2) The directions

with largest variance (high variation) are considered to be the most significant. In this case, one or a group of segments can only be well represented in the PCA space, while some segments could be considered as noisy data. For example, large-size segments, like the background segment that highly affects the variation of the data, could be well represented by PCA, while small size segments (which slightly affect the variation of the data) could be considered as noisy data. (3) If the data of segments are not linearly separable, PCA cannot well represent the segments; rather, PCA just represents them according to the variance of the variables, which could represent more than one segment in one cluster in the PCA space.

3.2.3 Optimal wavelength selection

HSI systems collect images with an enormous amount of spectral information in the form of a set of wavelengths. In fact, these systems are designed to cover a large area of applications such as meat processing, fruit processing and sorting and medical applications. Practically, many of these wavelengths are redundant and are not required to accomplish a particular task. For this reason, the selection of an optimal subset of wavelengths is an important and critical step in HSI analysis. There are three benefits of this step: (1) Minimizing the impact of collinearity between the variables (i.e., the wavelengths) [45], assuming that the collected spectra (wavelengths) of samples are independent variables. In this case, some of these independent variables could be highly correlated (collinearity), which could negatively affect the performance of a prediction model. (2) Selecting optimal wavelengths that represent the texture features of a material [62, 83]. (3) Improving the system in terms of robustness, accuracy and speeding up any proposed system to be valid for real-time applications [62].

In fact, there are many optimal wavelength selection methods proposed in the literature which aim to define the significance of each wavelength in solving particular classification or regression problems. These methods can be grouped into classical approaches and statistical methods.

Classical approaches

Classical approaches for optimal wavelength selection include regression coefficients of a prediction model [60, 78, 84, 85], PCA loadings [77, 82] and derivative spectra (1st or 2nd) [74, 79, 87]. The regression coefficients approach is based on fitting a supervised prediction model on the data such as a linear regression model or a *partial least squares* (PLS) regression model. Then, the peaks of the coefficients of the resulting model are used to select the significant wavelengths in the prediction of the model. These significant wavelengths are defined as high positive and high

negative coefficients, where each coefficient is related into a wavelength. Unlike the regression coefficients approach, PCA loadings and derivative spectra approaches are based on the nature of data (unsupervised analysis), while the significant wavelengths are extracted as the peaks of loadings and the peaks of derivative spectra.

The classical approach is commonly used in HSI data analysis due to its easiness in implementation. In [77], only six bands out of 237 bands were selected using the peaks of PC1 and PC4 loadings. These six bands were able to discriminate the lamb muscles (ST, LD, and PM). In [79], the second derivative was used to select those important wavelengths which have the needed information to detect and categorize the different types of red meat (lamb, beef, pork). The resulting five bands were used to build the same model again, the resulting model provided excellent results for solving this problem with just five wavelengths.

Statistical methods

Optimal wavelength selection using statistical methods includes *recursive feature elimination* (RFE) [88, 89], *successive projections algorithm* (SPA) [87, 91–93], *uninformative variable elimination* (UVE) [83, 87, 90] and *analysis of variance* (ANOVA) [94].

The RFE algorithm [88, 89] is a feature (variable or wavelengths) selection algorithm based on a search strategy to find the best subset of wavelengths that highly contributes to the prediction of a model. Iteratively, the algorithm starts with fitting a model by using all wavelengths. Then, the importance of each wavelength is computed as a rank for the wavelength. At each iteration, the wavelengths (the subset) that have the best ranks are extracted, then the model is refitted and its performance is evaluated and saved. Finally, the subset of wavelengths that has the best performance is determined, and the wavelengths in the subset are used to fit the final model for final evaluation. The RFE algorithm is considered to be an efficient algorithm for feature extraction due its simplicity. However, it has the following drawbacks: (1) The number of features in the subset needs to be defined. (2) The performance depends on the selected model to fit the data.

In the case of SPA, projection operations on wavelengths are used to select subsets of wavelengths that have minimum redundancy and collinearity. First, the algorithm starts with randomly selecting an initial wavelength and then projects this wavelength onto the remaining wavelengths. The wavelength with the maximum projection is selected as the candidate for optimal wavelength and then utilized as the new initial variable for the next iterations until the desired number of wavelengths is selected. Second, a prediction model is used to select the final optimal wavelengths from the candidate subsets; the final optimal wavelengths are selected based on the model's performance. It should be noted that the desired range of the

number of wavelengths needs to be defined for the algorithm [87, 91–93].

UVE is a wavelength selection method based on the regression coefficients of a PLS regression model. Unlike in the case of the classical regression coefficients approach, a noise matrix having the same dimension of the data is added to the original spectral data obtaining a new dataset. Then, two PLS regression models with cross-validation are fitted on the new dataset and original dataset. The resulting regression coefficients of the original dataset and the new dataset are obtained. To quantify the wavelength importance, an index defined as the ratio of mean value and standard deviation of regression coefficients is employed. Then, the highest index value of the new dataset is utilized as a cut-off threshold, where the wavelengths in the original dataset with an index value higher than or equal to the cut-off are extracted as optimal wavelengths [83, 87, 90].

The ANOVA method is defined as a collection of statistical models that could be used to analyse the difference between group means along with an associated methodology such as the variation among and between groups. Thus, analysing the effect of one wavelength at a time by the ANOVA method could provide useful descriptive information about the importance of each wavelength. However, this approach will not provide specific information about the relationship between variables and other important relationships in the entire data [94].

In general, these wavelengths selection methods are used for optimizing and customizing the HSI system for a specific application. Moreover, wavelengths selection methods are very important in the industrial calibration of an HSI system. For example, in [94], when an HSI system was implemented to detect the intramuscular fat regions in beef, only two wavelengths were selected by using the ANOVA technique; these wavelengths then were used to generate a ratio image for having the classification map. By using the same technique, the wavelength at $698\ nm$ has the best separation between the fat region and the meat regions. Thus, any proposed real application can only use this wavelength to improve the speed of prediction.

3.2.4 HSI image segmentation and feature extraction

In the general image segmentation framework, the main goal is to convert the HSI image from a pixel-oriented to an object-oriented structure (segments), where each segment (which has approximately the same spatial and spectral information) has a label and each label represents an object in the image [95, 96]. The spectral information is the reflectance intensities of the pixels, and the spatial information could be defined as the pixels' neighbouring relationships such as textural features.

The object-oriented structure of the image helps to deepen understanding of the contents of the image, especially in the case of heterogeneous scenes. Many algorithms have been introduced for image segmentation: thresholding, histogram-

based methods, *mean shift segmentation* (MSS), superpixel segmentation, edge detection, watershed transformation, fuzzy C-means method, region-growing method and split and merge methods [95,96].

In HSI image analysis, segmentation is used for extracting an accurate ROI of a sample and as an initial step for extracting the spectral and spatial features of an object in the image [60,62,77]. Due to the nature of HSI images, the segmentation methods are commonly applied on a single band (i.e., a single 2D grey image at a particular wavelength), a selected few bands [60, 63, 77] or a set of PCA score images [74].

Thresholding is a common segmentation method for HSI images in food processing applications. In thresholding, a proper threshold is selected for removing the background (i.e., unnecessary objects) and keeping only the ROI. Alternatively, multiple threshold values are selected as a multi-level segmentation for extracting multiple ROIs [60,63,74,77]. In fact, thresholding techniques need, and depend on, prior knowledge about the contents of images such as the number of objects or clusters, the proper threshold value by observing the spectra of materials or analysing the histogram of all wavelengths in the image [32].

On the other hand, advanced segmentation methods are adopted for HSI images in remote sensing applications such as the MSS algorithm [97] and superpixel segmentation [101]. In these methods, iteratively, the image is converted from pixel-oriented into segments based on certain criteria, where each segment shares the same spatial information and represents an object, or part of an object, in the image [97,101]. For HSI images, these methods are used for improving the accuracy of classification models [105–107]. These methods convert the HSI image into segments without any prior knowledge (i.e., human observation) about the contents of the images. However, they have tunable parameters such as the kernel size for MSS and the number and shape of the superpixels for superpixel segmentation algorithms [106,107].

In fact, HSI image segmentation is used as an essential step for extracting the spectral or spatial (texture) features of material in HSI images. For spectral features, the common approach is to extract the ROI by a segmentation method, then average the values of all the pixels in the extracted ROI having a single vector (spectrum) as the spectral feature of the material in the ROI. This approach is widely used for HSI imaging in food processing where the extracted spectra are used to represent the chemical composition of many food types [60,63,74,77].

The texture of meat (considered here as meat is the exemplar of this research) is formed by repeating units at different macro and microscopic scales, which are dependent on muscle type and the species being affected by the processing of the meat [11]. Thus, texture brings important information about meat. To have an ac-

curate measure of texture properties from HSI images, an accurate segmentation approach needs to be applied and then texture properties extracted for representing the spatial relationships between the pixels of a segment [11].

A common model for extracting texture features of an image is based on the spatial relationships of adjacent pixels by calculating how often a pair of pixels with the same intensity values occur in an image [95, 96]. This is estimated by using the *gray level co-occurrence matrix* (GLCM) model [95, 96]. Statistical texture features can be extracted from the GLCM, as proposed by Haralick [108], such as homogeneity, contrast, inverse difference moment, entropy, energy and correlation.

GLCM is typically applied to a single channel of an image and its use in hyperspectral data involves the estimation of features for each image in the hyperspectral cube or for selected images. It is also possible to use data-reduction methods such as PCA to reduce the spectral dimension and concentrate the information within a few images. For example, Naganathan et al., [109] investigated the use of hyperspectral image features for the classification of beef samples according to tenderness [109]. They used descriptive statistical features including Wavelet features, GLCM, Gabor features, Laws texture features and local binary pattern features. The features were extracted after reducing the dimensions of the hyperspectral images using PCA. The features extracted from the 2-day images were used to develop tenderness classification models for forecasting the 14-day beef tenderness. GLCM outperformed the other models and achieved a tenderness certification accuracy of 87.6%, an overall accuracy of 59.2% [109]. It should be noted that tenderness certification accuracy was defined as the ratio between the samples that correctly classified as tender-samples (true positives) and the total number of samples that predicted as tender-samples (true positives and false positives).

Bi-dimensional PCA (also called two-dimensional PCA [2DPCA]) is proposed as an extension of the original PCA by applying the same concept of PCA on 2D data structure such as RGB images [110]. It is proposed to directly obtain eigenvectors of the image covariance matrix without matrix-to-vector conversion. In this case, the size of the image covariance matrix is equal to the width of the images, which is quite small compared with the size of a covariance matrix in the case of the original PCA [110]. Thus, 2DPCA can analyse the image covariance matrix more accurately and compute the corresponding eigenvectors more efficiently than PCA [110]. In HSI images, 2DPCA can only be applied on each wavelength individually to extract textural or spatial information of samples at this wavelength.

Recently, Guo et al., [111] proposed the use of bi-dimensional PCA for the extraction of structural information and multi-features from hyperspectral data [110]. They observed that entropy values decrease with the increasing time of pork meat storage, where the higher value of entropy indicates a more uniform meat sur-

face [111]. It was proposed that changes in the surfaces indicated by changes in entropy could result from protein degradation that causes damage to the integrity of the structure of muscle cells [111]. In this case, the texture features of the hyperspectral image based on Gabor filters showed variations due to the storage time and mainly in the visible spectral range associated with myoglobin [111].

3.3 Hyperspectral data classification

The ultimate goal of many applications of HSI systems is to allow a decision process that requires a model to be able to translate the HSI data into information. This information can be grouped into a qualitative analysis (classification, e.g., yes/no decision) or quantitative analysis (regression, e.g., moisture content on the meat surface). In this research, we focus on the classification techniques. Typically, the classification techniques are grouped into unsupervised (input data are not labelled), semi-supervised (small amount of data are labelled) and supervised (all input data are labelled) classification. The use of these techniques depends on the availability of the source data. In HSI, there are two main approaches for HSI classification tasks: (1) Multivariate analysis and machine learning algorithms; commonly applied on HSI for food processing applications [19, 48, 55]. (2) Deep learning-based models; recently applied on HSI for remote sensing applications [57, 59, 112].

3.3.1 Machine learning-based approach

Recently, machine learning (or multivariate analysis, pattern recognition) algorithms have been considered *traditional* algorithms (also called in the literature *shallow learning algorithms*), due to the revolution of deep learning in the data science and computer vision fields. In these traditional algorithms, the features are hand crafted over 1D, 2D or 3D data domains. Then, the algorithms handle the features as 1D inputs for estimating their decision functions, which affects their performance in case of 2D or 3D data structures (i.e., 2D or 3D image formats); the performance depends on the kind of features and how the features are extracted.

In HSI for food processing, this approach (i.e., machine learning) is commonly used for a wide range of applications, e.g., prediction of quality [60, 61, 66] and safety [67, 79, 81, 82, 85] attributes of food products. Several machine learning models were proposed for classifying the HSI data of food. These models were grouped into: an unsupervised classification such as a PCA model for meat quality and safety classifications [74, 77]; and a supervised classification such as *linear discrimination analysis* (LDA), *partial least square discriminant analysis* (PLS–DA), *artificial neural networks* (ANN), *support vector machines* (SVM) and *k-nearest neighbour* (KNN). These

are the commonly used algorithms in the literature; however, there are other models that can be used for the same purposes. In this section, we focus on three machine learning algorithms that are commonly used in many applications in HSI for food processing: LDA, PLS–DA and SVM algorithms.

In fact, these algorithms were proposed for handling HSI data of food by two approaches: spectral features (as a 1D set of features) [60, 61, 66, 67, 79, 81, 82, 85] and combining spectral and spatial features(also as a 1D set of features) [62, 83]. In spectral features, after segmentation for extracting ROI, the mean spectrum is computed and then used as a representation of a sample (i.e., input of the model). In spectral and spatial features, the spectral features are extracted the same as in the first approach, while the spatial features are extracted, commonly, by GLCM or any texture analysis method, then these two kinds of features are concatenated as the representation of a sample [62, 83, 109, 111].

Linear discrimination analysis

The LDA algorithm is a classical machine learning algorithm [76]. It can be used for dimensionality reduction and multi-class classification tasks. In case of dimensionality reduction, the LDA algorithm aims to project the dataset into another space (usually lower-dimensional) while minimizing the variance within-class and maximizing the distance between the means of the classes. To quantify this objective, the algorithm computes the within-class scatter matrix (S_w) and between-class scatter matrix (S_b). The matrices can be computed as follows:

$$S_w = \sum_{i=1}^{C} \sum_{j=1}^{N_i} (\mathbf{X}_j^i - \mu_i)(\mathbf{X}_j^i - \mu_i)^\top \tag{3.10}$$

$$\mu_i = \frac{1}{N_i} \sum_{j=1}^{N_i} \mathbf{X}_j^i \tag{3.11}$$

where $\mathbf{X}_j^i$ is a vector defining the j-th observation from the i-th class, C is the number of classes, N is the number of samples in the i-th class, and μ_i is the mean vector of the class i. Similarly, the between-class scatter matrix can be computed as follows:

$$S_b = \sum_{i=1}^{C} N_i (\mu_i - \mu)(\mu_i - \mu)^\top \tag{3.12}$$

where μ is mean vector of the whole dataset.

Then, we solve the generalized eigenvalue problem of $S_w^{-1} S_b$ to obtain the linear discriminants. Similar as in PCA analysis, the eigenvectors with the highest eigenvalues carry the significant information about the distribution of the data. Thus,

the eigenvalues are sorted from highest to lowest, then the first K corresponding eigenvectors **B** are selected (as rows in matrix **B**). These eigenvectors can be used for transforming the input into the new space **Y** as follow:

$$\mathbf{Y} = \mathbf{B}\mathbf{X} \tag{3.13}$$

In case of classification, the LDA algorithm assumes that the distribution of the data is Gaussian. Thus, the LDA prediction function can be defined as follows:

$$DF_i(\mathbf{X}) = \mathbf{X} * (\frac{\mu_i}{\sigma^2}) - (\frac{\mu_i^2}{2 * \sigma^2}) + \ln(\frac{N_i}{N}) \tag{3.14}$$

$$\mu_i = \frac{1}{N_i} \sum_{j=1}^{N_i} \mathbf{X}_j^i \tag{3.15}$$

$$\sigma^2 = \frac{1}{N - C} \sum_{j=1}^{N} (\mathbf{X}_j - \mu)^2 \tag{3.16}$$

where σ^2 is the variance across the whole dataset **X**, N is the number of samples, N_i is the number of samples for class i, C is the number of classes, μ is the mean of the dataset, and μ_i is the mean of class i. DF_i is the discriminate function for class i given the input data **X**. Practically, for input samples, the outputs of all discrimination functions are computed, then the class of discrimination function that has the largest value will be the output classification process.

In fact, the LDA algorithm is frequently used for HSI classification in food processing applications [77, 80–82]. Mohammed et al., [77] used the LDA algorithm to investigate the ability of HSI systems to solve anatomical discrimination problems such as muscle discrimination. The average spectrum of three muscle types (ST, LD and PM) was computed and then the optimal set of wavelengths was estimated by PCA loading space, the selected wavelengths being used as input for the LDA model. The resulting LDA model achieved excellent accuracy compared with the two other models, conventional computer vision (i.e., by conventional RGB imaging) and the traditional chromameter method [77].

In [80], the LDA algorithm also achieved the best results compared with two classification methods: SVM and *soft independent modelling by class analogy* (SIMCA). This work implemented an LDA model for classifying the ripeness of persimmon fruit into four stages: unripe, mid-ripe, ripe and over-ripe. Two types of features were used: (1) The optimal wavelengths (518, 711 and 980 *nm*). (2) Texture features of the sample surfaces such as correlation, contrast, energy and homogeneity. However, in [82], for the same problem in [77], the LDA algorithm provided modest

results compared to some classification algorithms, such as ANN, SVM and *logistic regression* (LR). The comprehensive comparison in [82] between these algorithms showed that the ANN model achieved the best accuracy and is able to understand and analyse the spectral features more than the other algorithms [82].

Partial least square discriminant analysis

The PLS–DA algorithm is a supervised classification method based on the PLS approach[44]. The PLS-DA algorithm aims to sharpen the separation between groups of samples that define a class in the data. In fact, PLS–DA rotates PCA components to maximize the separation among classes and minimize the variation within classes. Thus, PLS–DA can be considered as a supervised version of PCA analysis. Moreover, the concept of PLS–DA is the same as for classical PLS regression. Unlike PLS regression, the output variable (**Y**) in PLS–DA is categorical variable expressing the class membership of the statistical units. Practically, the output $\mathbf{Y} \in \mathbb{R}^{N \times C}$ is represented as a dummy variable describing the classes, where N is the number of samples and C is the number of classes.

Similar to PCA analysis, the components of PLS-DA (also called *latent variables*) are linear combinations from the input data $\mathbf{X} \in \mathbb{R}^{N \times d}$ where d is the number of variables (wavelengths). The number of these components defines the dimension of the new space into which the original data will be transformed. Moreover, these components are orthogonal to each other. The components of PLS-DA can be defined as the eigenvectors of the non-singular portion of the covariance matrix **Cov**, given by:

$$\mathbf{Cov} = \frac{1}{(N-1)^2}\mathbf{X}^\top \mathbf{Cov}_N \mathbf{X}\mathbf{Y}\mathbf{Y}^\top \mathbf{Cov}_N \mathbf{X} \tag{3.17}$$

where N is the number of samples, $\mathbf{Cov}_N$ is $N \times N$ centering matrix, and **Y** is the output matrix containing the class labels.

In iterative processes, PLS–DA solves Eq.(3.17) and computes the eigenvectors a (transformation vectors), which give the importance of each feature in that component. In each iteration, the PLS-DA algorithm has the following objective function to be satisfied:

$$\max_{(a_i, b_i)} \mathbf{Cov}(\mathbf{X}_i, a_i, \mathbf{Y}_i, b_i) \tag{3.18}$$

where b_i is the eigenvectors for the matrix Y. X_i and Y_i are the residual (error) matrices after transforming with the previous $i-1$ components.

In fact, the PLS–DA produces the same output (i.e., class prediction) as the LDA algorithm, but the PLS–DA gives two advantages: noise reduction and a variable selection property [19, 45]. Mainly, the PLS–DA models provide two analytical tools:

latent variables (or PLS components) and regression coefficients. The number of latent variables influences the model complexity and stability. Usually, they are chosen using the cross-validation approach to avoid over-fitting or under-fitting in the model. Regression coefficients are used to select relevant variables (wavelengths) and reduce the dimensions of the HSI image [19, 45].

Also, the PLS–DA algorithm is widely used for classifying HSI data, especially in meat processing applications [78, 79, 81, 84, 92, 114]. In [84], the PLS–DA was investigated to classify pork meat into two groups: fresh and frozen-thawed. In this work, spectral features of 256 wavelengths, located in the NIR region ($900 \sim 1700$ *nm*), were used as inputs for the model. The complexity parameter of the model (number of components) was selected by using the *leave one out cross validation* (LOOCV) approach. As a comparison between the PLS–DA and LDA, Ropodi et al., [81] investigated these methods for studying the capability of multi-spectral imaging to detect the ratio of adulteration in red meat (pork and beef). The result showed that PLS–DA achieved excellent overall accuracy compared with an LDA model [81].

Support vector machines

The SVM algorithm [115] is a common machine learning algorithm for solving classification and regression problems. The SVM algorithm aims to find the best separation line (or hyperplane) between the data of two classes. The best hyperplane is defined as the hyperplane that has the largest margin from the nearest data points to the hyperplane. These data points are called support vectors. The main objective of the SVM algorithm is to find the optimal hyperplane which maximizes the margins between classes in the training data.

The SVM algorithm handles linear classification problems (linear SVM) and also non-linear problems (SVM with kernel functions). Practically, the SVM with kernels typically provides a better performance compared with the linear SVM. However, linear SVM models provide the variable selection property unlike kernel SVM models. In the case of the kernel SVM, the *radial basis function* (RBF) kernel is the usual standard and considered as a reference model. For the SVM with RBF kernels, there are two control parameters, the kernel radius and the cost or regularization parameter. These parameters are selected by using a two-step grid search with the cross-validation approach.

SVM algorithms are used in the applications of HSI of food [80, 82, 91, 92, 116]. In [91], a VIS-NIR HSI was implemented for detecting the freshness of prawn samples. The performance of three learning algorithms (SVM, AdaBoost and ANN) was evaluated on first and second derivative spectra of prawn samples; the results show that SVM with RBF kernel achieved the highest accuracy [91].

From a general review of the literature, it is concluded that there is no clear approach for deciding which of these machine learning algorithms defines an especially efficient algorithm either in general or for a particular problem addressed by these machine learning algorithms. This means that performance evaluation is application–dependent: it depends on the type of materials (spectral response), the type of selected features, the selected range of wavelengths and the relationships between the features. In general, the first recommended step is to investigate the linear models, such as LDA or PLS–DA, because it is easy to implement them, there are time and speed constraints and it performs well if the relationships in the spectra are linear (linearly separable).

The second step is to explore non-linear models like SVM with kernels, ANN and KNN. In [116], as an example, KNN achieved significant results compared with SVM in the applications of detecting foreign objects from wheat in HSI images. Finally, a comprehensive comparison between these types of models could provide us with a general overview for selecting the best algorithm and preprocessing technique for a particular application.

3.3.2 Deep learning-based approaches

Deep learning is defined as a class of models; these models learn deep features in a hierarchical way. The resulting features have a high level of abstraction, complexity and invariance to local change in the input data [117]. Recently, deep learning-based approaches have successfully provided highly accurate models in many research areas. Several models are designed for processing the data such as 1-D (like signals), 2-D (like the image structure), 3-D (like volumes of data such as HSI or MRI) and time series data.

Convolution neural networks (CNN) are considered to be robust tools due to their ability in dealing with many data types (i.e., 1-D, 2-D, or 3-D inputs). Thus, CNN models are commonly successful in vision-based tasks: image classification [118, 119], object detection [120], semantic segmentation [120], face recognition [121] and action recognition [122]. Moreover, CNN models achieve good performance in the field of signal analysis and speech recognition [123, 124].

Deep learning models yield a promising performance for solving HSI classification problems in HSI's many fields such as remote sensing [57, 59, 112], agricultural [125] and medical [126] applications. In general, many deep learning architectures have been proposed for handling HSI data of remote sensing such as *deep belief networks* (DBN) [127], *stacked autoencoder* (SAE) [128], CNN [125, 129–137], *recurrent neural networks* (RNN) [138] and *convolutional long short-term memory* (CLSTM) [139].

In [128], deep spectral and spatial features by SAE networks were proposed for

the first time in HSI imaging. SAE networks were used to extract deep features in an unsupervised way. The main goal of the SAE model was to convert the input into a new representation as learned features [128]. Then the extracted features were followed by an LR model for doing the classification part [128]. Three types of features were evaluated by this approach (i.e., SAE network and the LR model): spectral (1D vector), spatial (flatted $n \times n$ regions in PCA space) and joint spectral-spatial by stacking the two feature vectors as a one feature vector [128].

The same methodology as in [128] was implemented in [127] for extracting the learned features by using a DBN model with multiple *restricted boltzmann machine* layers. The whole architecture was trained by using a layer-wise training approach, then the resulting learned features were used as input for an LR classifier [127]. In fact, these methods (i.e., SAE, LSTM and DBN), as in [127,128], achieved a high performance compared with the shallow machine learning algorithms, such as SVM, KNN and ANN. However, they still do not consider the spatial information as a 2D structure, where converting the $n \times n$ regions into a 1D vector destroys the meaning of spatial information and the benefit of having the HSI data as an image.

Recently, CNN networks have been proposed as adaptive tools for HSI classification in remote sensing applications [57,59,112]. CNN networks showed a kind of robustness for handling the spectral features (as 1D-CNN networks) [129], spatial features (as 2D–CNN) [130] and joint features of both (as 3D–CNN) [130]. In [129], a CNN model was proposed for the first time for HSI classification, a 1D vector (pixel value or spectrum) was processed through several layers: 1D-CNN, max-pooling, fully connected and output layers. This implementation showed efficiency in comparison with a set of reference machine learning models, although the spatial features were totally ignored [129].

Inspired by [129], Chen et al., [130] proposed 2D–CNN and 3D–CNN models for utilizing the spectral and spatial features of an HSI image. For the 2D–CNN model, a window of $(N \times N)$ in the PCA space was used as a representation of the input and then passed through three 2D–CNN layers with 2D kernel representation and two pooling layers [130]. For the 3D–CNN model, a spatial window of $(N \times N \times \lambda)$ was used as input of three 3D–CNN layers with 3D-kernels and two pooling layers [130].

The comparison between these approaches showed that the 3D–CNN model is better in terms of accuracy and deep understanding of HSI data [130], whereas the 1D-CNN approach, as in [129], ignores the spatial features, and the 2D–CNN approach depends on a preprocessing step (i.e., PCA transformation) and the 2D–CNN layers perform the convolution over each channel individually (i.e., the 2D kernels) [130]. Thus, the 3D–CNN approach has the advantage of extracting joint spectral-spatial features in a single operation [130].

An optimized 3D–CNN model was proposed in [131]. The model consists of two

3D–CNNs and a fully-connected layer, in addition to the input (as a 3D window) and the output layer [131]. In this model, a small number of feature maps, small kernel size and small spatial size, were investigated and showed efficiency in terms of accuracy and training speed (light-weighted model) [131].

Alternatively, the ability of CNN networks to understand the relationships of neighbouring pixels was investigated in [133]. A number of CNN models, depending on the number of neighbouring pixels, were implemented for classifying a pair of pixels, then the predictions of all models were postprocessed by a majority voting strategy [133]. Moreover, CNN networks showed robustness in HSI image reconstruction [132], where the CNN layers in this work were used to emphasize the similarity between the classes inside the HSI image [132].

In the last two years, CNN networks have received more research attention for classifying HSI image data. In addition to the above works, several models were proposed based on combining different CNN structures such as: 3D–CNN with 1D-CNN [134] or 3D–CNN with 2D–CNN [135, 136]; combining multiple 3D–CNN layers by a multi-scale approach [137]; and combining CNN layers with other deep learning models such as CNN with RNN layers [138, 139]. In these approaches, the main goal is to enforce the deep learning models to have a high level of input data abstraction and deep and complex sets of features for classification.

In [137], a multi-scale 3D–CNN model was investigated for having an abundant context of HSI data. For achieving multi-scale property, 3D–CNN blocks were proposed, where each block has several 3D–CNN layers with different kernel representations. Then, in a layer-wise approach, the resulting feature maps were combined by sum operations and passed to the next 3D–CNN block [137].

In fact, these approaches (i.e., deep learning models) are considered to be state of the art in HSI classification for remote sensing applications due to their efficiency compared with the shallow machine learning algorithms. The efficiency of deep learning in the context of HSI classification can be summarised as follows: (1) The performance of deep learning models against machine learning algorithms on standard datasets (i.e., publicly available datasets). (2) Unlike machine learning models, deep learning models can extract useful information directly from the image without destroying the shape of the data, which increases the benefit of having the data as 2D or 3D structure-like images. (3) Deep learning models can process the whole image data in a single operation (like a 3D–CNN operation). (4) Deep learning models can extract learned features in a hierarchical structure from the raw input data, while machine learning algorithms use handcrafted features by preprocessing the input image data. However, deep learning models show challenges in terms of the need for a large number of samples, selecting a particular architecture for a particular problem, optimizing the optimal parameters of the architecture and training the

architecture.

For HSI of food processing, there is a gap in the literature, and deep learning models for this kind of data have not been sufficiently examined. Thus, the implementation of the deep learning model for solving food processing problems is an active research area, where more consideration about the effects of variation between pixels (effects of light source distribution) and between samples (effects of chemical and biological composition of samples) and the possibility of real-time models need to be taken into account.

3.4 Summary

Recently, HSI systems have received more attention in many applications in food processing research areas such as meat processing, fruit processing and other agricultural applications. This large scale of application makes HSI systems efficient tools due to their robustness in providing both spatial and spectral distributions of food products. Moreover, HSI systems are non-destructive tools in the testing phase of modelling.

In the literature, studies utilize HSI data often as follows: using handcrafted spectral features by a spectral preprocessing method or using the same spectral features and handcrafted texture features. These kinds of features are used as input (as a 1D vector) for a shallow machine learning algorithm for classification or regression problems. These approaches provide success and good results. However, they have the following drawbacks (based on our opinion): (1) They need strong preprocessing such as preparing the samples by experts and extracting handcrafted features for spectral or spatial information. (2) The used spatial features depend on the methods that were used to extract these features, which reflects on the model's accuracy, where the types of methods and spatial features need to be optimized for having good accuracy. (3) The used features (for spectral or spatial) for the classification model are extracted individually and in the structure of one-dimensional data, which reduces the benefit of having the data as a 2D structure like images.

Besides the success of deep learning in computer vision research, the motivation of using deep learning in HSI classification is for the following reasons: (1) It can deal with complex situations of lighting conditions in the imaging, for example, a pixel from the same class appears as different spectral information in different locations. (2) It can handle the whole image data in a single operation (like a 3D–CNN operation) without destroying the shape of the data, which increases the benefit of having the data as 2D or 3D structure-like images. (3) It can extract learned features in a hierarchical structure from the raw input data (no need for strong prepro-

cessing methods to be applied). Thus, deep learning models can generate a high level of spectral or spectral-spatial data abstraction, which is invariant to different conditions like scattering in lighting, spectral variation, sample variation and sample moving and rotation. Moreover, deep learning is much faster than traditional methods in the testing step, while it is very slow in the training step. In [130], the classification time of the HSI image using CNN networks is evaluated, where the proposed CNN model achieves a very good classification time compared with other algorithms like the SVM model. *Graphic processing units* were suggested to improve the classification time of the proposed model [130]. Thus, deep learning models could be appropriate for real application by HSI systems.

Given these advantages of the deep learning approach and that these kinds of models are not sufficiently studied in files of HSI for food processing, there is a good opportunity to fill the research gap and put more research effort into deep learning models for analysing the chemical and textural distributions of materials. In fact, several existing HSI deep learning models (including CNN and other models) were proposed for remote sensing applications and they outperformed the state-of-the-art models. These deep learning models were optimized and evaluated for classifying a single image, where there is no variation between images (i.e., the resulting variation by capturing many HSI images and the chemical variation by changing samples as in food processing application). Moreover, in remote sensing applications, the classification is a general material classification, not a fine-grain material classification, where most of the classes are visually different. Thus, for other kinds of applications such as food processing, there is a need for more investigation and novel approaches for handling the variation of many images, the lighting conditions and the limitations in spectra of snapshot HSI images. All of these considerations need to be addressed while implementing an accurate and complex deep learning model for food processing applications.

Chapter 4

Spectral and Textural Features for HSI Image

This chapter provides a comprehensive analysis of the performance of hyperspectral imaging (i.e., line scanning hyperspectral imaging) for handling the interaction of both chemical distribution (i.e., spectral) and textural (i.e., spatial) properties of red meat. Moreover, it presents a novel deep learning framework for extracting and combining these properties in a single prediction model by using CNN networks.

We use the detection of adulteration in red meat as a case study for testing and evaluating the proposed framework, while the same framework can be fitted on any type of food products. For simulating the adulteration problem, meat muscles were defined as either a class of lamb or a class of beef and pork. A dataset of line scanning images of lamb, beef or pork muscles was collected, taking into account the state of the meat (fresh, frozen, thawed and packing and unpacking the sample with a transparent bag).

We investigated the impact of spectral features of red meat by using PLS–DA and SVM models, taking into account several spectral feature extraction methods. Also, we investigated handcrafted spectral and spatial features by using the SVM model and self-extraction spectral and spatial features by using a deep CNN model. Results showed that the CNN model achieves the best performance with a 94.4% overall classification accuracy independent of the state of the products.

The rest of this chapter is organized as follows. Section 4.1 provides an introduction which include the motivations and objectives of the research presented in this chapter. Section 4.2 reviews the state-of-the-art techniques which are related to this research. Section 4.3 describes the used dataset, HSI system and the developed models and methods. Section 4.4 presents the experimental setup and results of the proposed framework. Result analysis and discussion are given in Section 4.5. Section 4.6 summarizes this chapter.

4.1 Introduction

Recently, HSI systems, as illustrated and defined in Chapter 2, have gained attention in a plethora of research areas such as medical applications [31], remote sensing

imagery [30, 56–59] and food and meat processing [27, 32, 48, 51]. The key advantages of HSI systems are that they facilitate the visualization of materials inside the image, the distribution of their chemical components, and the texture of their spatial distribution.

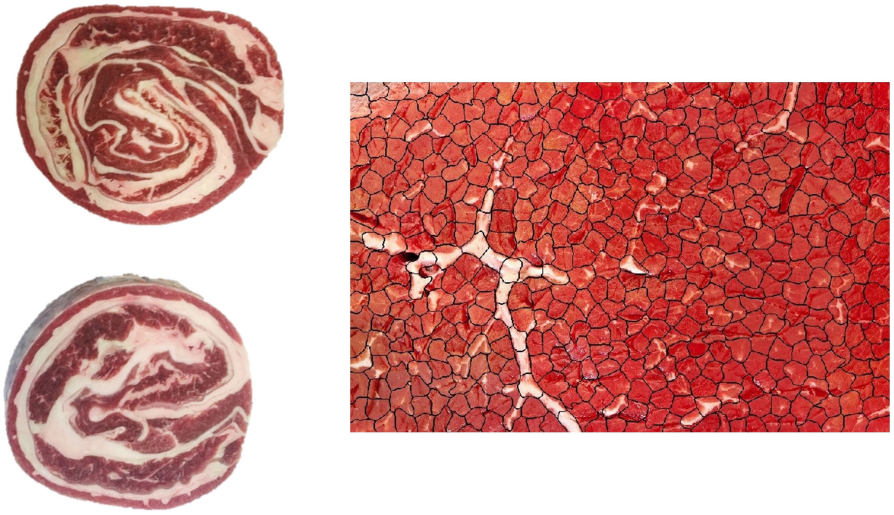


Figure 4.1: *Left*: Example of rolled meat products. *Right*: A representation of the repeated textural units of a meat sample.

Regarding meat, the exemplar of this research, HSI spectral information is used for the assessment of many chemical attributes such as PH-value [60–62], water-holding-capacity [63], intramuscular fat [94], tenderness [65–67] or springiness [68]. In these studies, only spectral features were used as a representation of the chemical components of meat, while the spatial structure of samples was ignored.

The texture of meat is formed by repeating units at different macro and microscopic scales which are dependent on muscle type and species being affected by processing the meat. Thus, the texture brings important information about the meat and significantly improves the prediction models for grading the quality and safety of meat products [11, 45]. Figure 4.1, right, shows examples of these repeated units, where each unit shares the same local textural properties of meat at particular spatial locations. Thus, all units provide a deep description and global understanding of the whole meat sample which logically reflects on the performance of a prediction model.

Meat processing is generally evaluated in terms of quality and safety attributes. HSI systems are used to assess both attributes and lead to good results in terms of accuracy and applicability, compared with traditional methods (i.e., lab-based and spectroscopy methods) [19]. A meat-quality evaluation approach provides a scaling factor indicating the quality of the meat at the time of taking the HSI image (i.e., quantitative indicator). A safety-evaluation approach aims to detect unusual artefacts that may have been added to or had accrued in the meat sample [45].

One of the critical safety-based evaluation methods of meat products is the detection of adulteration of meat products, for example, the addition of another type of meat which may have a lower price compared to the original material. From an industry point of view, this type of adulteration requires greater attention [33–38, 38], for example, by detecting any adulteration in pre-packed rolled meat products (Figure 4.1, left, shows an example of these products).

HSI offers the possibility of detecting the unique characteristics of meat both spectrally and structurally (i.e., based on texture). But, based on our knowledge, there are still no approaches that are able to extract these two types of characteristics within a single modelling framework. Thus, the aim of the research, presented in this chapter, is to develop a novel approach that combines both spectral and textural information into a single model. The specific objectives of this chapter are as follows:

- Investigate the influence of practical conditions on the spectral response of red meat. Considered conditions are: (1) packing meat into a transparent bag; (2) freezing meat for three days; and (3) thawing meat after being frozen.
- Develop a classification model to discriminate one type of meat muscle from the others (e.g., in the case of adulteration), for example, identify lamb meat that is different to beef or pork with the conditions mentioned above.
- Investigate the impact of HSI spectral features for discriminating lamb meat from the other types by the state-of-the-art classification models.
- Develop a methodology to extract joint features (i.e., handcrafted spectral and textural features) for HSI images applied to this type of application.
- Investigate the applicability of deep learning approaches and their robustness on this type of application (i.e., meat-processing applications).
- Develop and evaluate a novel deep learning model for self-feature extraction, by combining spectral and textural properties of meat in a single model of HSI images of red meat products (i.e., by handling the raw input instead of handcrafted feature extraction).

4.2 Related work

Traditionally, quality and safety attributes of food are evaluated either by spectroscopic tools [9, 10] or traditional colour images [26, 28]. The studies described in [26,28] show applications of colour images for the assessment of food quality. Recently, HSI systems have been considered as robust tools for processing and evaluating food products, for example, classifying or predicting attributes related to meat quality and safety.

In [77], for lamb muscle discrimination, the results showed that the performance of HSI systems are more efficient than other optical systems like RGB images and chromameter devices (a colour meter device); by chromameter, three colour values from the visible region were measured for each sample and then averaged to obtain a single measurement for each sample. Also, HSI systems were investigated in [79] for red meat classification tasks by employing the PLS–DA model on the second derivative spectra of red meat. The results showed good accuracy in the case of sample-based prediction, but the model achieved low accuracy in the case of pixel-based prediction [79]. Thus, the authors proposed a majority-voting mechanism to obtain final classification of the whole image [79].

A simulation of the red meat adulteration problem was presented in [140]. In [140], two algorithms, PLS–DA and SIMCA, were tested for discriminating the lamb muscle from other red meat (i.e., beef or pork). The results show that the PLS–DA model outperforms the SIMCA model, but the performance of the PLS–DA model was unstable and is dependent on the way samples were presented [140] (i.e., vacuum packed or without packaging). Also, the results in [64, 81, 82] showed that an HSI system is able to provide significant information for performing classification in a plurality of applications for meat such as detection of adulteration of minced meat [81], detection of chicken adulteration in minced beef [64] and lamb muscle discrimination [82]. In all of these studies, the models produced misclassification of pixels in pixel-based prediction, although they performed well in the case of sample-based prediction. Reasons for the misclassified pixels include the construction of models by using the average spectrum over the whole sample and ignorance of the spatial variation in the pixel space. In fact, the source light strongly affects the pixels in the acquired HSI image (light scattering or illumination effects).

In the case of heterogeneous samples, classifying HSI data is a real challenge in terms of covering the variation of spectral and spatial information in the sample and pixel space, respectively. As an example, the detection of any adulteration is seen in the case of pre-packed rolled meat products (an example is shown in Figure 4.1, left). In this case, it is more practical and reliable to perform a pixel-wise classification (i.e., local) than a sample-wise classification. For dealing with spectral variations,

several methods were established and used, such as spectral derivatives, SNV and MSC. Considering texture features is still a gap so there is room for more research in the field of HSI data classification [11].

The basic strategy for classifying hyperspectral data is to treat the contribution of each pixel (i.e., the spectrum signature) as a sample. This strategy is usually applied to HSI for remote sensing applications [105] due to the limitation in images. Thus, it provides a powerful model which takes the local variation in the image into account. In HSI for meat processing [64, 79, 79, 81, 82, 140], the used strategy is averaging all pixels in the ROI as a spectral signature of the ROI. In this case, the resulting models considered only the spectral features to be used, while the spatial features were ignored. So, the results were useful but not fully satisfactory, especially in the case of heterogeneous samples.

In [141], a method for joint feature extraction was proposed by using superpixel segmentation. *Simple linear iterative clustering* (SLIC) algorithm [101] was employed to produce superpixels of HSI images. Then, the mean of each super-pixel was calculated and used as input for an SVM model. Then, the resulting SVM decision values were postprocessed by using a linear conditional random-field model to give a class label for each superpixel. In general, the use of superpixels in HSI classification tasks adds the following advantages: (1) Stability of the extracted signatures by averaging the superpixel at each wavelength [141]. (2) The possibility of exploring the neighbourhood relationship between highly correlated pixels [107].

In [107], three types of features were evaluated through a multi-kernel composition of an SVM-RBF: raw spectrum; the average of each superpixel; and the weighted average of neighbours for each superpixel. In both [107, 141], the results show that superpixels enhance the fitted models. However, this consideration may badly affect the fitted model if the extracted superpixels are inaccurate. So, the application of ensemble-based classification methods was recommended [107].

Recently, numerous deep learning models have been proposed as classification and self-feature extraction models for analysing HSI data such as deep CNN models [125, 129, 130, 139]. In [129], a CNN model was introduced for the first time for HSI remote sensing data. Five layers of deep learning architecture were proposed and structured as follows: spectrum signature (i.e., pixel vector) as an input layer; two 1D CNN layers with a max-pooling operation; and one fully-connected layer followed by an output layer. The proposed deep CNN models have many parameters (to be optimized), such as the number and length of hidden layers or the size of filters of the convolution operations. However, the results show that the 1D CNN model is able to understand and extract a non-linear relationship of the samples of each class and outperforms the traditional models for classifying HSI data such as SVM or MLP networks. For the achieved results, the model was only considered

for the spectral features; the spatial features were ignored. Practically, standard 2D CNNs, such as in [118], are not applicable for HSI due to their nature (i.e., high-dimensionality problems) [130]. In [130], deeper and more complex CNN models were evaluated on HSI for remote sensing. In addition to a 1D CNN model, a 2D CNN was proposed and extended to be applicable for HSI application. The PCA was applied to reduce the dimensionality of the input (i.e., the depth of the hyper-cube) for three channels (i.e., the highest three components); a window around each pixel was used as input to the model. Then, the input was passed through two convolution layers, a pooling layer, a fully-connected layer and an output layer.

The 2D CNN model was extended into a 3D CNN model [130, 139]. In the case of the 3D CNN, 3D convolution kernels were used to extract joint features across the whole dimensions of the input; this means that there is no need for the PCA to pre-process the input. The evaluation results show that the 3D CNN achieved optimal performance, but has more parameters (like the size of the 3D window around the target pixel) to be optimized [130,139]; more samples are required to train the model, where the window size was the most effective parameter. The extracted features by 3D convolutions are inspired by the 3D texture features which take the spectral and spatial domains into account [130, 139]. Moreover, the features are invariant to the effects of local edges in the case of pixel-based classification [139].

4.3 Materials and methods

4.3.1 Hyperspectral imaging system

As this research aims to understand and utilize the interaction of both chemical (i.e., spectral) and textural (i.e., spatial) distributions of red meat, we find that a line-scanning HSI system is the tool which is commonly used for predicting many of the chemical attributes of red meat. Line scanning HSI systems usually provide the spectra of hundreds of wavelengths at a particular wavelength range. The system for acquiring HSI images, used in this chapter, is described and demonstrated in more detail in Section 2.4.1 where all the system's setup, parameters and reflectance calibration are explained.

4.3.2 Dataset and sample preparation

Meat and fat samples were procured from three local butcher shops and two different local supermarkets in New Zealand. A total number of 75 samples, including lamb (18), beef (13), pork (13), and fat (31), were prepared by cutting into squares of size $2.7 \times 2.7 \times 1.1$ *cm* and then drying with normal tissue. Then, the samples were

labelled and kept at 2° C for 16 hours. The next day, the samples were taken from the fridge and put into well-designed containers (frames shaped as a matrix of meat and fat species; Figure 4.2, left, shows examples of these frames).

Then, each frame was scanned with the HSI camera (meat and fat were in a fresh state). Then, all frames were vacuum packed into a transparent bag and re-scanned with the HSI system (in a fresh state with packing). After that, all frames were frozen (-4° C for three days). After three days, the same frames were re-scanned by using the HSI camera (frozen state with packing). Then, the packing was removed and HSI images were collected for the frames (frozen state without packing). Finally, the frames were left at room temperature for three hours for thawing; then, the frames were re-scanned by the HSI system.

As shown in Figure 4.2, the total number of prepared frames is six frames: four frames [including 57 samples of lamb (12), beef (9), pork (9) and fat (27)] were used for training purposes and one special frame for testing purposes (18 samples of lamb (6), beef (4), pork (4) and fat (4)). In the frames, the meat and fat samples preserved their original meat textural characteristics, similar to retail-ready products commonly found at supermarkets. The total number of collected HSI images is 25 including: 20 images used for training purposes and five images for testing and evaluation. All images were calibrated based on the method as described in Section 2.4.1

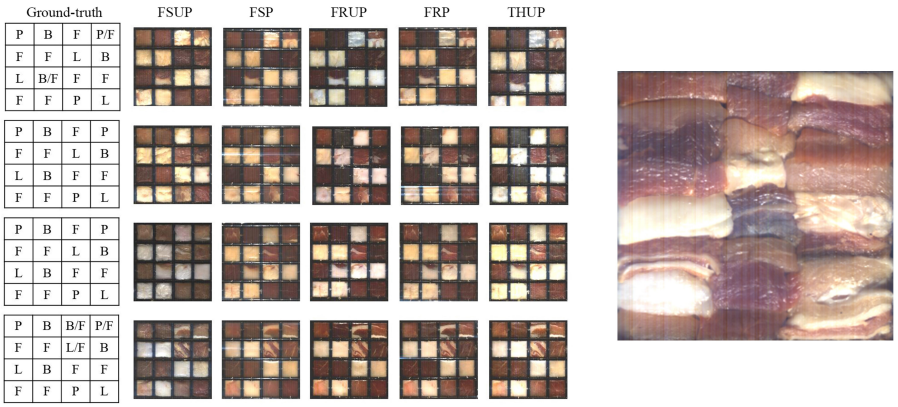


Figure 4.2: *Left*: The calibration samples in frames: *ground truth* (GT) and false-colour images for frames $1-4$ (top to bottom), where FSUP is short for fresh red meat unpacked, FSP for fresh packed, FRUP for frozen unpacked, FRP for frozen packed, and THUP for frozen-thawed unpacked. In the cells of GT, F is for fat, L for lamb, B for beef, and P for pork. *Right*: Evaluation-sample frame in case of FSUP.

4.3.3 Spectral data analysis and visualization

The adulteration in red meat products is simulated by defining the following classes of labels: lamb meat is labelled as one class (called *LAMB*), another class for both beef and pork (called *OTHER*) and a class *FAT* for visualization purposes. By defining this structure, the model predicts the probability of lamb meat against the others. This situation is more practical and reliable from an industry point of view. Moreover, we evaluate the power of HSI images to discriminate one material from any other predefined materials.

When dealing with HSI data, multivariate analysis methods are needed in high-dimensional space. This high dimensionality prevents the visualization and pattern investigation steps during the analysis of the data. However, PCA is considered as an appropriate model for HSI image data in terms of data distribution visualization and reducing the dimensions of the data (i.e., extracting the most important information) [79, 82].

The dataset that is described in Section 4.3.2 was used for estimating a PCA model. The calibration set of images was manually segmented (according to the GT) into small regions (approximately 100 pixels); then, the mean spectrum of each segment was extracted. A PCA model was fitted for these spectra. The loadings of the PCA model were then used for transforming each spectrum into the PCA space (PCA scores) by projecting the original spectrum values on each loading vector; the number of loading vectors is equal to the number of wavelengths and equal to 225, where these loading vectors can be used to approximate the original data.

The first five vectors present the original data with an accumulated explained variance of 98.42%. Thus, these five vectors were selected for further analysis and the rest were considered as vectors presenting noisy data. Thus, projecting the spectra (225 dimensions) on these five vectors results in an approximation of the spectra with only 5 dimensions (PCA scores). In fact, these scores were used in this research for the following purposes: (1) For visualizing the patterns between the pre-defined classes (LAMB and OTHER). (2) As a preprocessing step for extracting the spatial features.

Figure 4.3, left, shows the mean spectrum of the extracted spectra for each class; it clearly shows that they are highly correlated in shapes and there are significant differences in the reflectance of mean spectrum for each class. Figure 4.3, right, visualizes the class separation (meat types and their status) in the PCA space. It should be noted that the first and fourth PCA component were empirically selected to visualize the spectra and class separation of each meat type and status, where these two components present the original data with an accumulated explained variance of 97.67%.

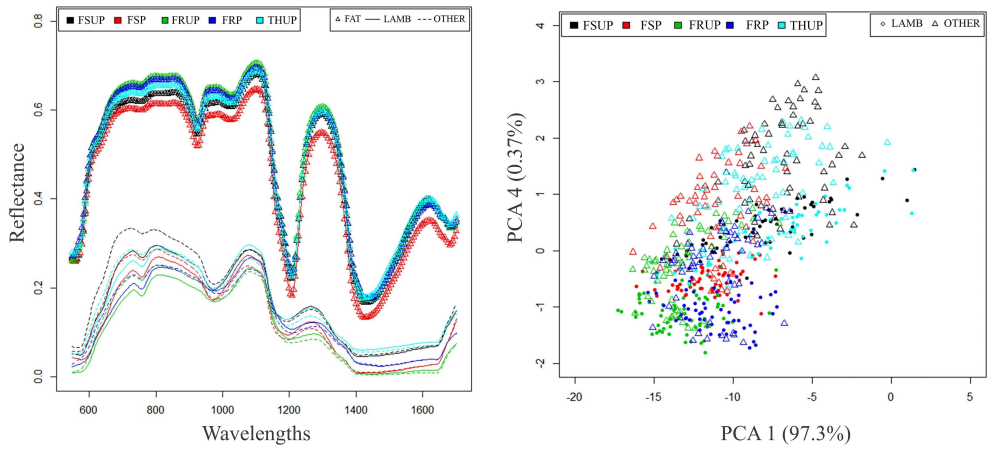


Figure 4.3: PCA analysis. *Left*: Mean spectrum of each of the three classes for each of the five statuses. *Right*: Scatter plot for the LAMB and OTHER classes, showing 97.67% of the total variances of the original data.

Empirically, we found that these two components provide the best visualization for highlighting the patterns between meat types and their status, while the second and third components provided patterns between meat and fat, or fat at different statuses. The class distribution, as shown in Figure 4.3, right, shows that the classes are separated into regions with overlapping regions. This overlapping is due to the frozen status of the meat, which is expected as freezing affects the physicochemical properties of meat [86]. Overall these results illustrate the ability of spectral information to discriminate the multiple sources of information in the data (type of meat, status froze x fresh).

4.3.4 Model-based classification framework

In general, the selection of each pixel in an image as a sample is impractical in terms of computation resources and time due to the large number of those pixels. Also, hand-segmentation for sampling each class is inefficient because the local features of the segments are not accurate in this case. For these reasons, we proposed superpixel segmentation as a sampling method to collect representative samples (i.e., pixels or segments) for each class from the dataset. In fact, the pixels in each superpixel share the same local spectral and spatial features that reflect on the performance and stability of any fitting model.

Hyperspectral image segmentation

Superpixel segmentation is considered as a convenient approach for capturing local features of images and a preprocessing method to reduce the complexity of complicated image segmentation tasks in terms of processing time and quality of resulting segments. Thus, the superpixel approach has become useful for a wide range of applications in computer vision such as image segmentation [98], body model estimation [99] and object localization [100].

In hyperspectral classification tasks, the advantages of using superpixels can be summarized as follow: (1) They provide more representative spectral and spatial information than pixels. (2) Pixels in a superpixel share similar spectral and visual properties which means that superpixels have a perceptual meaning. (3) They provide a sufficient and compact spatial representation of images which can be used for sampling images into set of representative local features.

In this research, the SLIC superpixel algorithm [101] is adapted to generate the superpixels of HSI images. In fact, the SLIC algorithm was selected to be used in this research after investigating the literature. For example, other algorithms like quick shift [102] and graph-based segmentation as in [103] do not provide a parameter to control the number of resulting superpixels [101]. Also, the computation time is a big challenge in HSI images; results in [101] demonstrated that SLIC performed better than turbopixels [104] and quick shift algorithms in case of big size images. Moreover, SLIC was successfully applied and provided good results in different tasks in HSI [107, 141].

The SLIC algorithm was originally proposed for conventional images (i.e., three channel RGB images or RGB images in LAB space). For HSI images, the algorithm is not directly applicable due to the high dimensionality of images. However, in this study, the SLIC algorithm is adapted to be used for HSI images by reducing the dimensionality of images by the PCA approach.

First, the estimated PCA model, as described in Section 4.3.3, is used to reduce the dimensionality of input images and then extract the first five-score images. These score images are obtained by projecting the input images onto the first five loading vectors (also called eigenvectors or bias vectors) of the PCA model, where these loadings represent the most significant information from the input HSI image (the five loadings have an accumulated explained variance of 99.1%). Then, the score images are combined to form a new 3D image (PCA image) with a shape of $(N_{row}, N_{col}, 5)$, then, the PCA image is used as input to the SLIC algorithm. Figure 4.4, b, shows an example of these PCA images resulting from an input HSI image shown in Figure 4.4, a. Second, the adapted SLIC algorithm for HSI images computes superpixels by clustering image pixels based on their similarity in

the PCA space and proximity in the image plane. Thus, the clustering is done in 7-dimensional space, PCA1 to PCA5 approximating the spectral information (pixel values) and xy defining the pixel location.

For an input image with N ($N = N_{row} \times N_{col}$) pixels and S, superpixels need to be generated; the algorithm considers an approximate size for each superpixel equal to N/S. Thus, the superpixel centres could be located at every grid interval $W_s = \sqrt{N/S}$. Assuming that the possible spatial extent of a superpixel is approximately its area W_s^2, then all pixels that are associated with its cluster centre lie within an area of $2W_s \times 2W_s$ around the superpixel centre on the image plane. The window $2W_s \times 2W_s$ in the xy plane is the search window for finding pixels nearest to the cluster centre.

For computing the distance (similarity) in the 7-dimensional space, between pixel i and cluster centre k, an adapted Euclidean distance is used and computed as follows:

$$d = d_{PCA} + \frac{m}{W_s}\, d_{xy} \quad , \quad \text{where} \tag{4.1}$$

$$d_{PCA} = \sqrt{\sum_{n=1}^{5} (PCA_{n,k} - PCA_{n,i})^2} \tag{4.2}$$

$$d_{xy} = \sqrt{(x_k - x_i)^2 + (y_k - y_i)^2} \tag{4.3}$$

where d is the sum of the distance in the PCA space and the xy plane distance is normalized by the grid interval W_s. The variable m is a tunable parameter controlling the compactness of a superpixel.

The cluster centers are initialized first by selecting center pixels at regular grid steps W_s in the image plane. Then, the centers are updated by using the gradients of the image to avoid placing them at an edge and to reduce the chances of selecting noisy pixels. The algorithm iteratively assigns a segment label for each pixel in the image and updates the cluster center vectors, where each label identifies a cluster or superpixel in the image. A residual error as L1 distance is used to stop the process (algorithm convergence). Then, an enforced connectivity process is applied to relabel disjoint segments with the labels of the largest neighbouring cluster. Algorithm 12 shows the processes of generating superpixels of an HSI image.

Algorithm 1: Superpixel segmentation of an HSI image

Input : HSI image, S, m
Output: Cluster centers, segmentation labels

Apply PCA on the input image
Extract the first 5 PCA images
Initialize cluster centers $Cl_k = [PCA1_k, PCA2_k, PCA2_k, PCA2_k, PCA2_k, x_k; y_k]$ by selecting pixels at regular grid steps W_s in the image plane
Update cluster centers in an 3×3 window, to the lowest gradient position in the image
repeat
    foreach *Cluster center* Cl_k ***in*** Cl **do**
        Assign label to the best matching pixels from a $2W_s \times 2W_s$ around the cluster center according to Eq. (4.1)
    end
    Compute new cluster centers by averaging the pixel values (mean vector in the 7 dimensional space)
    Compute a residual error E (L1 distance between previous centers and new centers)
until $E < threshold$
Enforce connectivity

Extraction of spectral features

The resultant SLIC-segmented labels, as described in the above sub-section 4.3.4, are utilized to extract the spectra of each class. In fact, due to the computation costs and class balancing, representative pixels from each superpixel need to be extracted. The *Kennard stones* (KS) algorithm [142] was proposed for dividing datasets into two subsets in a systematic approach: calibration and validation. The main advantage of the KS algorithm over random sampling is that KS approach guarantees that all validation samples fall inside the distribution of the calibration samples [142]. Thus, the selected calibration or validation samples could be considered as representative samples of the whole dataset.

First, the algorithm starts searching for finding the best two samples that are the farthest apart from each other in data space; the algorithm uses PCA to reduce the dimensionality of the data (in case of high dimensional data) and then uses the Euclidean distance in PCA space for finding these two samples. These two samples are assigned to a calibration dataset and then removed from the original dataset.

This search procedure is repeated until reaching the desired samples to be selected as representative samples; the representative samples could be for any purposes like training (calibration), validation, testing or visualising.

So, we use the KS algorithm [142] to sample each superpixel into a subset of representative pixels (raw spectral features in reflectance) as final samples for modelling. After extracting the raw features, we investigate the impact of the chemical distribution of meat, presented by the spectra of meat, on classification models by using different spectral feature extraction methods. These methods are: *spectral derivatives* (1st or 2nd), *SNV normalization* (SNV–norm) and spectral L_2 *normalization* (L_2–norm).

In case of spectral derivatives, the SG derivative [69] is used for extracting the 1st and 2nd spectral derivatives with polynomial fitting for maintaining the shape of the spectra. These techniques remove the effect of additive or multiplicative noise from the raw spectra, and they aim to highlight the local patterns of spectral signal at a particular set of contiguous wavelengths.

Spectral normalization methods aim to highlight the global patterns of a single spectrum by re-projection of the spectrum into a shared scale or shared statistical properties (e.g., shared mean and variance). The proposed spectral normalization methods are defined as follows: Let $P(x,y) = [u_1, u_2, ..., u_n]^\top$ be a pixel in an HSI image at location (x,y). The L_2–norm and SNV–norm of $P(x,y)$ are computed by using Eqs. (4.4) and (4.5), respectively:

$$P^{\circ}(x,y) = \frac{P(x,y)}{||P(x,y)||_2} = \left[\frac{u_1}{||P(x,y)||_2}, \frac{u_2}{||P(x,y)||_2}, ..., \frac{u_n}{||P(x,y)||_2}\right]^\top \tag{4.4}$$

$$\overline{P}(x,y) = \left[\frac{u_1(x,y) - \mu_P}{\sigma_P}, \frac{u_2(x,y) - \mu_P}{\sigma_P}, \ldots, \frac{u_n(x,y) - \mu_P}{\sigma_P}\right]^\top \tag{4.5}$$

where $||P(x,y)||_2 = \sqrt{{u_1}^2 + {u_2}^2 + ... + {u_n}^2}$, μ is the mean of the target pixel P and it is computed as $\mu = \frac{1}{n}\sum_{i=1}^{n} u_i$, and σ is the standard deviation of the target pixel P and it is computed as $\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(u_i - \mu)^2}$.

Practically, we found that these normalization techniques are very sensitive to spike points. The spike points are abnormal behavior caused by the camera detector at a specific wavelength, resulting in a sharp rise in the spectrum followed by a sharp decline at this wavelength. Also, the spike points spike points can result from specular points on a sample's surface which could saturate the detector at a specific wavelength or group of wavelengths; the specular points are usually accrued in HSI systems due to the effects of the lighting system. To reduce the impact

of these spike points in data analysis, we used SG filtering [69] as a preprocessing step for smoothing the raw spectra and for removing these spike points in spectra. In this case, SG filtering uses the neighbours of a spike point to predict new values, taking a multi-order polynomial fitting into consideration while estimating the new value; neighbours are defined as a window of odd elements centred around the spike point. Thus, the corrected values of the spike points are interpolated values from their neighbours with the advantage of taking the shape of spectrum (i.e., the polynomial fitting) into account.

Extraction of spatial and textural features

In fact, hyperspectral cameras are established for a wide range of applications, such as fruit sorting, medical applications and meat processing. They provide an enormous amount of spectral information, represented as a group of gray-level images for each HSI image. So, the extraction of spatial features of an object inside an HSI image is a real challenge; especially the decision as to which gray-level image (or images) is more useful in order to extract the spatial features.

For dealing with this challenge, we adapt RFE [88, 89] and *random forest* (RF) [89, 143] algorithms to select the most significant wavelengths (i.e., bands or variables) for discriminating between the red meat muscles. From our PCA as in Section 4.3.3, we observed that the spectral response of each class is strongly influenced by the status of the meat (i.e., the overlapping regions between the classes as shown in Figure 4.3, right). For this reason, we propose a meat status-based methodology to select a set of bands to be used as a reference image for the spatial feature extraction process. Thus, five independent RFE algorithms were implemented one for each status, then the most significant wavelength from each RFE was selected to be used for further analysis the spatial feature extraction process. It should be noted that in the case of frozen meat status, two wavelengths were selected for this status. From the results of the RFE algorithms, six wavelengths were selected, which have the highest importance at each status (optimal wavelengths). The resulting wavelengths (all in *nm*) are as follows:

optimal wavelengths : $636.6,\ 646.5,\ 656.3,\ 932.3,\ 1134.4,$ and 1154.1

The spectra of LAMB and OTHER classes and their labels, as described in Section 4.3.2, were used in the implementation of the RFE algorithms; the spectra of FAT class were not considered in this analysis. The dataset was divided into five datasets based on the status of meat (i.e., FSUP, FSP, FRUP, FRP and THUP). For each status, the dataset was randomly separated into train and test sets. Then, an RF model was fitted by using the train set with all wavelengths (i.e., 225 wavelengths)

to classify the spectra into LAMB or OTHER classes. Based on the RF model, the importance of each wavelength was calculated. To calculate the importance, the prediction accuracy of the RF model on the test set is recorded. Then, the same is done after permuting a wavelength. The difference between the two accuracies are then averaged over all models and normalized by the standard error, this difference in accuracies defines the importance of this wavelength.

Algorithm 2: The proposed RFE algorithm to select optimal wavelengths from an HSI dataset

Input : Datasets (one for each meat status)
Sub-sets of wavelengths $S = \{1, 2, 4, 8, 16, 32, 64, 128\}$
Output: Set of optimal wavelengths

```
foreach dataset in Datasets do
    2 Randomly divide the dataset into training (70%) and testing (30%) sets.
    Train and tune an RF model on the training set using all wavelengths.
    Predict the testing set of samples using the trained RF model.
    Calculate the importance of each wavelength based on the RF model.
    foreach s in S do
        Keep only s wavelengths that have the most importance.
        Train and tune an RF model on the training set using s wavelengths.
        Predict the testing set of samples using the trained RF model.
    end
    Calculate the performance of models over all elements in S.
    Select the best sub-set that have the highest performance.
    Train and tune an RF model on the training set using only the selected
     sub-set wavelengths.
    Calculate the importance of each wavelength in the selected sub-set.
    Select the wavelength that has the highest importance (optimal
     wavelength).
end
```

The RFE algorithm is a search algorithm for estimating the importance of a sub-set of wavelengths based on a performance of the RF fitted model. Thus, a list $S = \{1, 2, 4, 8, 16, 32, 64, 128\}$ defines the number of wavelengths (sub-set of wavelengths) that need to be selected. Then, for each sub-set, the algorithm keeps only the sub-set that has the most importance and excludes the rest. Then, an RF model is trained by using this sub-set of wavelengths and evaluated by the test set of samples. This process is repeated for all sub-sets in S. The best sub-set is selected based on the out-of-sample accuracy of the models on the test set of samples. Finally, the

selected sub-set is used to fit a new RF model, then the importance of each wavelength for this sub-set is calculated and sorted. The wavelength that has the highest importance is selected as the final optimal wavelength to present the dataset (i.e., the status). Algorithm 16 shows a pseudo-code demonstrating the proposed RFE processes to select the most significant wavelengths to be used for spatial features extraction.

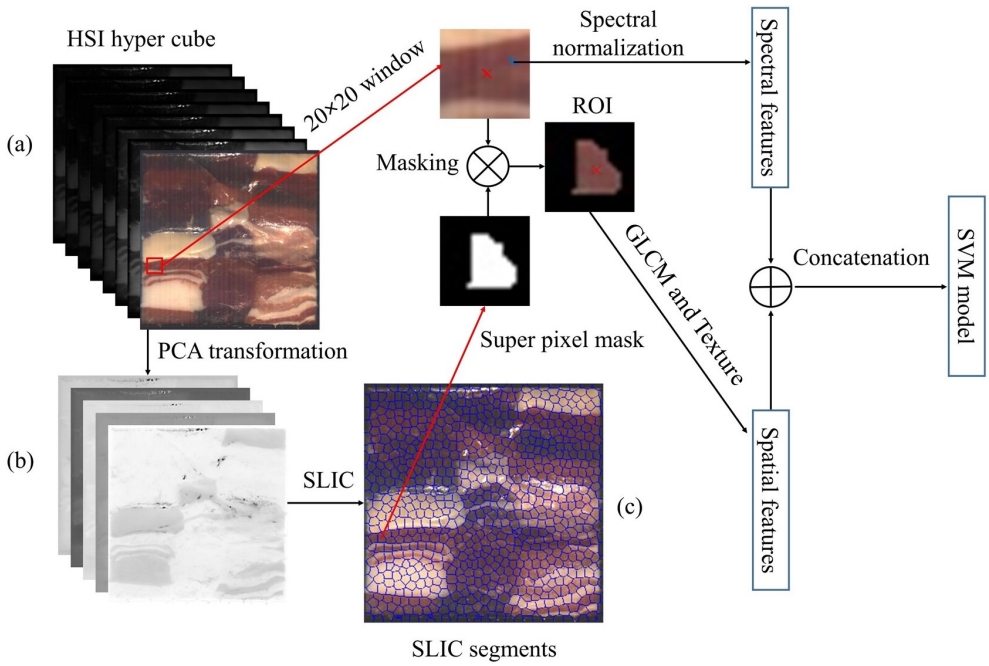


Figure 4.4: A representation of ROI extraction from an HSI image for extracting the texture and spectral features. (a) A heterogeneous hyperspectral image of meat muscles. (b) The first five score images of the HSI image. (c) Superpixel segments of the HSI image.

Texture-based models are generally used in computer vision as the spatial features. The GLCM is one of the standard ways for extracting the textural properties of a set of neighbouring pixels, defined by a window around the target pixel. GLCM is computed by calculating how often a pair of pixels with the same intensity values occur in an image. In addition, the moving distance and angle between the target pixel and the others need to be defined.

A set of statistical features were proposed in [108] so as to define the spatial relationships between the neighbouring pixels (textural properties). These features are computed from the GLCM matrix. Thus, we propose the following textural features for defining the texture of red meat muscles: *homogeneity*, *contrast*, *entropy*, *energy* and *correlation*. These features are computed as follows:

$$Homogeneity = \sum_{i,j=0}^{N-1} \frac{p_{i,j}}{1+(i-j)^2} \tag{4.6}$$

$$Contrast = \sum_{i,j=0}^{N-1} p_{i,j}(i-j)^2 \tag{4.7}$$

$$Entropy = \sum_{i,j=0}^{N-1} p_{i,j}(-\ln p_{i,j}) \tag{4.8}$$

$$Energy = \sum_{i,j=0}^{N-1} \frac{p_{i,j}}{1+(i-j)^2} \tag{4.9}$$

$$Correlation = \sum_{i,j=0}^{N-1} p_{i,j} \frac{(i-\mu)(j-\mu)}{\sigma^2} \tag{4.10}$$

where $p_{i,j}$ is an element of the normalized symmetrical GLCM, N is the number of gray levels (256), μ is the GLCM mean, and σ is the standard deviation of the intensities of all reference pixels in the relationships that contributed to the GLCM matrix.

In the case of HSI images, the texture features are not directly extractable due to the nature of the HSI image (hyper-cube). However, the optimized gray-level bands (i.e., the resulting six optimal wavelengths) were used to compute the GLCM matrix and then extract the mentioned textural features. As our application is a heterogeneous sample, we utilized the superpixel method to extract *masked textural features*. The developed feature extraction (masked textural features) method is described as a pseudo-code as shown in Algorithm 3, and visually demonstrated in Figure 4.4.

In fact, the masked textural features allow the model to be robust with regard to the heterogeneous sample (it avoids the overlapping between muscles) and provide more accurate texture representation (considering large window size while computing the texture).

Algorithm 3: Spectral and textural features extraction from an HSI image

Result: Joint spectral and textural feature of an HSI image
Input : HSI image
Output: Features matrix **X**

Compute PCA score images of the HSI input image
Mask the input image to extract only optimal wavelengths (*nm*) $[636, 646, 656, 932, 1134, 1154]$ as set of gray images.
Normalize the optimal wavelengths into a range of $1 - 255$.
Compute superpixels of the input image using SLIC segmentation and the first five PCA score images.
foreach *superpixel* ***in*** *superpixels* **do**
    Compute the centroid of the superpixel
    Crop a 20×20 window around the centroid
    Mask the selected window by the superpixel mask to obtain the ROI
    foreach *optimal wavelength* ***in*** *the selected window* **do**
        Compute the GLCM matrix with a distance of 1 and direction of 0
        Eliminate the first row and column of the resulting GLCM matrix
        Compute homogeneity by Eq. (4.6)
        Compute contrast by Eq. (4.7)
        Compute entropy by Eq. (4.8)
        Compute energy by Eq. (4.9)
        Compute correlation by Eq. (4.10)
    end
    Select set of representative pixels from superpixel by KS method: 11 for LAMB, 9 for OTHER, and 9 for the FAT class.
    foreach *pixel* ***in*** *pixels* **do**
        Normalize the spectral features by Eq. (4.4) or Eq. (4.5)
        Concatenate the texture and spectral features in one feature vector
        Save the final feature vectors in **X**
    end
end

Machine learning models for meat processing

In the literature, the state-of-the-art studies utilize machine learning models for HSI data of meat using spectral information and handcrafted texture information. This

information is then presented as input (as a 1D vector) for a machine learning model. The SVM and PLS-DA models are considered the most successful for classifying HSI data of meat. These approaches provide good results. However, they have the following drawbacks: (1) They need strong preprocessing for extracting handcrafted features for spectral or spatial information. (2) The used features (spectral or spatial) for the classification model are extracted individually and in the structure of one-dimensional data, which reduces the benefit of having the data as a 2D structure like images.

In this section, spectral-based features of meat and the joint features of meat texture, as well as its spectral features, are proposed to be extracted. To evaluate the proposed features, we use the commonly used machine leaning models for classifying HSI data of meat: SVM with RBF kernel as a non-linear model and PLS–DA as a linear model; following the state-of-the-art studies for classifying HSI data of meat. In the case of spectral-based features, we evaluate both SVM and PLS–DA models to obtain the best spectral features. In the case of joint features (i.e., spectral and texture features), we use only the SVM model for performance evaluation (a decision based on observations from our experiments).

Deep learning approaches are considered state of the art for many tasks in computer vision research, such as for image classification. As in this research we use HSI images (3-dimensional data structure) and the pixel-wise classification approach, the deep learning models for RGB images and the models for image classification (i.e., classifying the whole image into a category) are not comparable with the research literature reviewed in this thesis. Moreover, several existing HSI deep learning models (including CNN and other models) were proposed for remote sensing applications [57,59,112] and they outperformed the state-of-the-art models.

However, these deep learning models for remote sensing applications [57,59,112] are not comparable with the research proposed in this thesis for the following reasons: (1) These deep learning models were optimized and evaluated for classifying a single HSI image as a dataset, where there is an assumption that the lighting conditions are fixed. (2) In remote sensing applications, the classification is a general material classification, not a fine-grain material classification, where most of the classes are visually different. (3) In these studies, the models were adapted and changed for each dataset (i.e., remote sensing HSI image), which means that the models perform well only on a specific type of HSI data.

Thus, applying these approaches on HSI data for meat (our research) is not practical for the following reasons: (1) In the collected data set of HSI images, the lighting conditions are variable because the images were collected during multiple experiments. (2) There is a variation in the data by capturing HSI images of different meat samples. (3) The classification between the meat types is fine-grain classification,

where the classes are visually similar.

In light of the above, for meat processing applications, there is a need for more investigation and novel deep learning approaches to be evaluated against the traditional machine learning model, such as the SVM model. At the time that this research was started, there were no deep learning models for classifying HSI data of food or meat. Thus, in the next section, a novel deep learning approach is proposed and comprehensively evaluated against the SVM model.

4.3.5 Deep learning-based classification framework

The main objective of this thesis is to investigate methods for addressing the interaction of textural and spectral information of meat which are provided by HSI systems. This objective inspires us to investigate a multi-input deep learning architecture, where each input is a sub-architecture enabling the learning of specific kinds of features from the input HSI image. To make these features dependent and combined, shared layers need to be added to connect the output of these sub-architectures together and with the output of the whole architecture. Moreover, multi-input deep learning models provide robustness in terms of considering the different shape, nature and representation of the samples, while different types of information are shared in one deep model.

In this section, we introduce a novel deep learning approach for detecting the adulteration in red-meat products (the same approach could be applied for any type of food product). Based on our best knowledge, this is the first time that deep learning methods have been applied to HSI imaging for meat processing. The performance of the proposed model will be compared with several models with handcrafted spectral and spatial features. The proposed deep learning model is multi-input deep learning architecture. In fact, the proposed model aims to cross both the spectral and spatial domains for extracting self-features of an HSI image. In the next sections, we explain the basics of the proposed deep learning architecture.

Architecture of the proposed multi-input CNN model

CNN models are mostly utilized as a 2D CNN in computer vision such as image classification and recognition. However, CNNs are successfully used as 1D CNN in signal processing such as speech recognition and noise filtering. In these cases, the input of 1D CNN is illustrated as a vector (or $n \times 1$ array). Thus, the dimensionality of this spectral signature of HSI data is applied to CNN classification models [129]. Also, 3D CNN models are proposed for handling the temporal features of video sequences in a time series, for example, in action recognition tasks [121]. The HSI

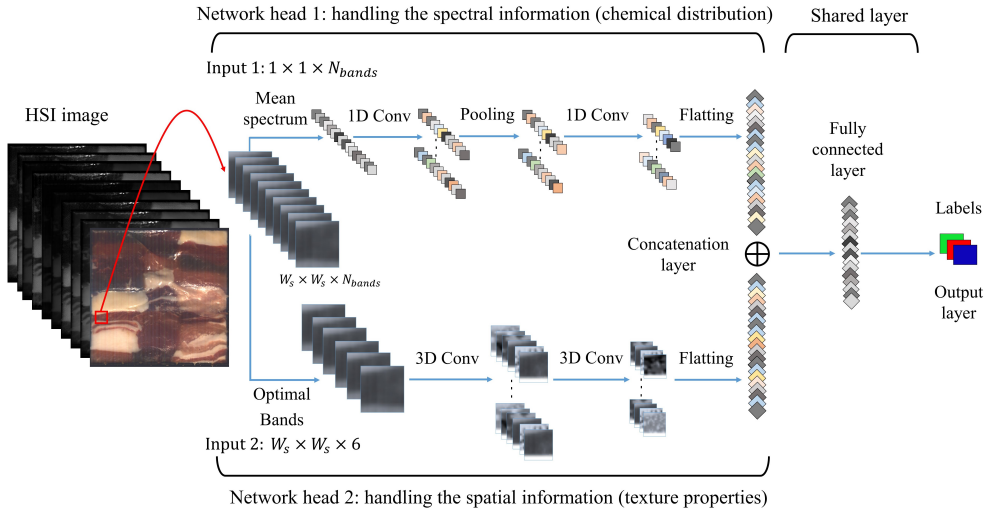


Figure 4.5: An illustration of the proposed multi-input CNN model. Input 1 represents the mean spectrum as a vector. Input 2 shows the spatial window at the selected optimal bands as a 3D cube. The labels represent the predefined classes (i.e., LAMB, OTHER, FAT).

hyper-cube can be illustrated as a sequence of spectral 2D bands. So, the 3D convolution operations are able to extract joint features across the spatial and spectral domains of an HSI image data [125, 130, 131].

Thus, deep CNN models show robustness and flexibility in handling different data structures such as 1D, 2D or 3D data structures. The flexibility of the CNN models motivated the research to use CNN models to extract useful features from an HSI image of meat. These features can be defined as follows: (1) Local spectral features by analysing the mean spectrum of neighbours of the target pixel. As the spectrum is 1D data structure, we propose 1D CNN layers to extract the spectral features of the target pixel. (2) Local spatial features by analysing the region around the target pixel. For the spatial features, we propose 3D CNN layers as the region around the target pixel is 3D data structure. (3) Global features as a combination of the local features.

This inspired us to propose a multi-input CNN architecture which consists of two sub-architectures: an architecture for processing and analysing the spectral features of meat in the image(called *network head* 1); and an architecture for the spatial features of meat in the image (called *network head* 2). For extracting global features,

the extracted features from each sub-architecture are concatenated and then connected with a shared layer (fully-connected layer) as a global set of features crossing the two types of information (i.e., spectral and spatial information). Figure 4.5 demonstrates these sub-architectures and the shared global feature extractor (i.e., the shared fully-connected layer).

The number of layers in each sub-architecture was empirically selected based on a cross-validation evaluation using the training data, where each sub-architecture was trained and evaluated independently. It should be noted that this evaluation is only for estimating the best number of layers in each sub-architecture, where the other parameters at this stage are fixed. Based on this initial evaluation, we empirically found that only two CNN layers for each sub-architecture are the optimal values for our dataset. Also, we found that adding pooling layers is only useful in the case of network head 1.

In light of the above, the proposed multi-input CNN model consists of a hierarchical structure of layers: two input layers, convolution layers, a down sampling layer (pooling layer), a concatenation layer, a shared layer (fully-connected layer), and an output layer (classification layer). Figure 4.5 shows the structure and the combination between these layers. In all CNN and fully-connected layers, we use *rectified linear units* (ReLU) for activating the output of these layers. While, the *softmax* function is used for computing the output layer.

The extracted mean spectrum, represented in Figure 4.5 as input 1, has a size of $(1, N_{band})$ and feeds forward to the first 1D convolution operation with K_1 kernel size, a stride size of 1, and M_1 feature maps with a size of $(M_1 \times N'_{bands} \times 1)$ where

$$N'_{bands} = |N_{bands} - K_1 + 1| \tag{4.11}$$

The max-pooling layer (1D max-pooling operations with pooling size of 2 and stride size of 2) is used for downsampling the spectral response into a new size. Thus, the size of feature maps is reduced into a size of $(M_1 \times N''_{bands} \times 1)$, where

$$N''_{bands} = |N'_{bands}/2| \tag{4.12}$$

The next convolution layer has a K_2 kernel size and stride size of 2, which produces M_2 feature maps with a size of $(N'''_{bands} \times 1)$, where

$$N'''_{bands} = |(N''_{bands} - K_2 + 1)/2| \tag{4.13}$$

Then, these nodes are flattened into one vector as the first local feature with a size of $(M_2 \times N'''_{bands} \times 1)$.

The second input is defined as a 3D fixed window around the target pixel, then masked with only the selected optimal wavelengths (in Section 4.3.4). This input

layer is shaped as $(W_s, W_s, 6)$, where 6 is the number of optimal bands, which will be passed into two 3D convolution layers. In this case, the convolution operations are carried out by using a 3D kernel. So, the first convolution layer has $\overline{M}_1$ kernels of size $(\overline{K}_1^1 \times \overline{K}_1^2 \times \overline{K}_1^3)$. Then, the resultant maps are passed to the next 3D CNN layer in the same way by kernels of size $(\overline{K}_2^1 \times \overline{K}_2^2 \times \overline{K}_2^3)$ and generate another set of feature maps $\overline{M}_2$. In these 3D CNN layers, we pad the input with the same value and a stride size of 1 was used, so the output of these layers is the same as the input layer (i.e., the second input layer of the model). Similar to the first input, the final features are flattened as a vector having dimensions of $(\overline{M}_2 \times W_s \times W_s \times 6)$.

The local features from these two networks are then connected with a fully-connected layer (i.e., the shared layer). This layer extracts a high level of features from spectral and spatial inputs (global features), then the classification layer makes the final prediction based on these global features.

Training strategy of the proposed multi-input CNN model

Training the model aims to adjust the model weights based on an error loss between the actual output and the model prediction (model output). Thus, the training procedure consists of two main processes: Forward propagation and back propagation. The forward process takes each input layer and passes it through the convolution layers. Thus, the input of $i-th$ convolution layer is computed as follows:

$$X_i = f(v_{i-1}) \tag{4.14}$$

where $v_{i-1} = W_{i-1}^{\top} X_{i-1} + \mathbf{b}_{i-1}$, and $W_{i-1}^{\top}$ and $\mathbf{b}_{i-1}$ are the weight matrix and the bias vector of the previous layer. For non-linear transformation, $f_{(.)}$ is utilized as an activation function for these convolution layers. Thus, the ReLU activation, which is used in the proposed architecture, is defined as follows:

$$f(v) = \max(0, v) \tag{4.15}$$

where v is any real. Then, the features (i.e., the global features) are concatenated and passed to the next fully-connected layer with the same process. The output layer of the model is defined as the softmax layer; the softmax activation function produces an output with a probability distribution over the predefined classes.

In the back-propagation process, the weights of this model are updated by using the standard mini-batch *stochastic gradient descent* (SGD) algorithm. SGD adjusts the model weights by minimizing a loss function between the actual output (GT) and the model output. In the proposed model, we used the *categorical cross entropy* function as the loss function of this model, and it is computed as follows:

$$\mathcal{L}(\theta) = -\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{C}\{j = Y^{(i)}\}\log(y^{(i)}) \tag{4.16}$$

where θ is the model parameters, n is the number of samples, Y is the GT vector (encoded as one hot vector style), y is the model prediction vector, and C is the number of classes.

The SGD algorithm iteratively adjusts the weights of the model over many iterations (i.e., epochs), which means the model sees the same samples many times. In fact, this process could produce an overfitting problem in the model. In this case, the model performs well on the training samples but poorly on the test samples. To avoid the overfitting problem, several regularization techniques are proposed in the literature, such as the dropout, L_1, and L_2 regularization. In the proposed CNN model, we adapt the L_2 regularization as a generalization method. L_2 regularization encourages the model weights to be small by adding a penalization term to loss function in Eq. (4.15). The final loss function is defined as:

$$\mathcal{L}(\theta) = -\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{C}\{j = Y^{(i)}\}\log(y^{(i)}) + \frac{\ell}{2n}\sum_{k=1}^{m}W_k^2 \tag{4.17}$$

where m is the number of weights W and ℓ is the tunable parameter.

4.4 Experiments and results

We used the dataset, as described in Section 4.3.2, for obtaining the results of the proposed methods. The proposed features of model-based methods were evaluated by using PLS–DA and SVM classifiers. For models assessment, we used a 10-fold cross-validation with grid search for hyper-parameter selection. PLS–DA, SVM, and RFE methods were implemented based on caret package, an R-analysis tool for classification and regression models. Also, the results of the proposed deep learning model were obtained based on KERAS API, a python library for high-level deep learning programming.

For evaluating the proposed methods, we used the F_1 score and the *overall accuracy* (O.A) as measures to evaluate the performance of each method. The F_1 score shows the accuracy of each class by combining both precision and the recall which provides a harmonic mean of both. F_1 and overall accuracy are computed as follows:

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{4.18}$$

$$O.A = \frac{\sum tp + \sum tn}{\sum number\ of\ samples} \tag{4.19}$$

where the recall is defined for each class as $tp/(tp+fn)$ and the precision as $tp/(tp+fp)$; $tp, tn, fp, and\ fn$ are true positive, true negative, false positive and false negative values of a confusion matrix, respectively.

For models evaluation, a set of GT images were manually labelled by using Scyven software; Scyven is software for HSI images analysis and exploration. Figure 4.6 shows examples of these GT images. In the evaluation of all models, we extract all labelled pixels from each GT and the resulting classification map for obtaining the accuracy of each meat status. For measuring the final accuracy of the models, as independent to the state of meat, we average the resulting measurements from the previous step.

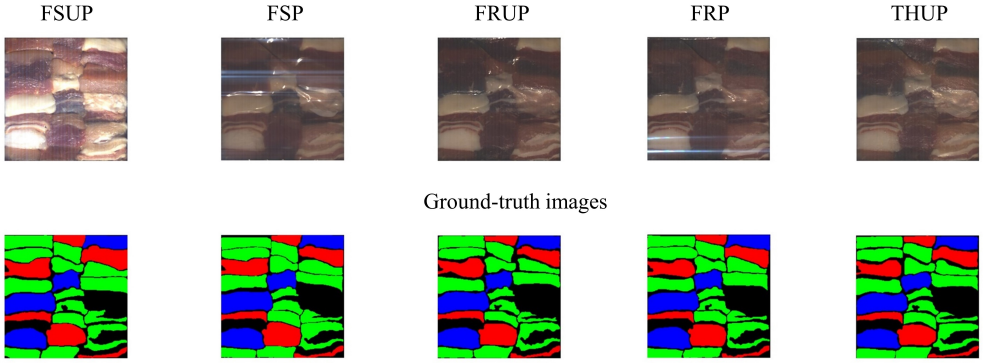


Figure 4.6: Ground-truth images of the testing HSI dataset. *Top*: False-colour images of HSI images at each meat status. *Down*: The GT images. The colours Red, Green and Blue represent classes LAMB, OTHER (beef or pork) and FAT, respectively.

4.4.1 Model-based classification framework

The estimated PCA model, as in Section 4.3.3, was used to transform the original HSI image into its PCA space (i.e., score images). Then, the first five scores are selected and used for generating the superpixels. The region size of each pixel was empirically set to 100 pixels (i.e., 10×10). The pixels in each superpixel were re-sampled into a limited number of pixels as follows: 11 for LAMB, 9 for OTHER, and 9 for the FAT class. The KS algorithm was used for re-sampling each superpixel into a group of representative pixels.

Spectral-based classification

The selected pixels of each class were smoothed across the spectral domain by using the SG method. Empirically, we chose 9 and 2 for the window size and the order of the polynomial function of the SG algorithm, respectively. Then, the following spectral feature vectors were obtained (the length of these vectors is 225): (1) The raw spectra. (2) 1st derivative spectra by SG method with window of 9 and 2nd order polynomial fitting. (3) 2nd derivative spectra by SG method with same setting as in 1st derivative spectra. (4) The L_2–norm by using Eq. (4.4). (5) The SNV-norm by Eq. (4.5).

Table 4.1: Performance evaluations of both SVM and PLS–DA models on the proposed spectral feature vectors on average of all meat status.

Model	**Feature vector**	**F_1 score**			**O.A**	**F_1 (avg)**
		LAMB	OTHER	FAT		
SVM	Raw spectral	80.3	86.7	96.6	87.3	87.9
	1st derivative	81.7	87.3	97.4	88.6	88.8
	2nd derivative	79.9	83.7	96.7	87.3	86.7
	L_2–norm	80.2	86.2	98.0	87.2	88.1
	SNV–norm	86.4	91.1	97.6	91.4	91.7
PLS–DA	Raw spectral	69.9	79.6	77.2	75.3	75.6
	1st derivative	70.6	79.9	77.7	76.4	76.1
	2nd derivative	70.5	78.9	78.1	75.8	75.8
	L_2–norm	71.4	80.2	80.1	77.4	77.2
	SNV–norm	74.8	83.3	83.2	80.4	80.4

Table 4.1 shows the results of the PLS–DA and SVM models of the proposed spectral features (on average of all meat statuses); the results of each meat status are also available in Appendix A as Table A.1. The results in Table 4.1 clearly show that the SVM model performs better than PLS–DA model in all feature vectors, which reflects non-linear patterns in the data of each meat status. In addition, the spectral normalization techniques significantly achieve better results than raw and derivative methods. Based on these results, we chose the SVM model as the best model and spectral normalization as the best features for red meat. Thus, we used them for the texture extraction and analysis.

Texture-based classification

As described in Section 4.3.4, the textural features were extracted and then concatenated with the spectral vector of the target pixel. By using the SVM model, we investigated the effect of adding the textural features into different types of spectral features (i.e., the joint features of spatial and spectral features). Thus, the investigated feature vectors are as follows: raw-spectral with textures (raw-texture), L_2 normalized spectral with texture (L_2–norm–texture) and SNV normalized spectral with texture (SNV–norm–texture). Table 4.2 shows the influence, on each model's performance, of adding these textural properties into the spectral response of the muscle types.

Table 4.2: Performance evaluations of the proposed combination of spectral feature vectors and textural features over each meat status by using SVM model.

Meat status	**Feature vector**	**F_1 score**			**O.A**
		LAMB	OTHER	FAT	
FSUP	Raw-texture	88.4	93.7	98.1	93.1
	L_2-norm-texture	88.9	94.2	99.1	93.9
	SNV-norm-texture	91.4	95.6	98.8	95.3
FSP	Raw-texture	76.2	82.3	97.7	85.1
	L_2-norm-texture	80.2	87.5	97.9	87.9
	SNV-norm-texture	84.1	90.6	97.6	90.3
FRUP	Raw-texture	79.1	85.2	98.6	86.9
	L_2-norm-texture	79.2	86.1	97.8	87.1
	SNV-norm-texture	86.9	92.5	98.5	92.5
FRP	Raw-texture	81.5	87.4	97.3	88.5
	L_2-norm-texture	81.4	89.1	96.2	88.9
	SNV-norm-texture	83.3	90.3	97.1	90.2
THUP	Raw-texture	81.8	87.4	98.1	89.2
	L_2-norm-texture	85.1	91.3	98.6	91.4
	SNV-norm-texture	91.9	95.7	98.7	95.5

4.4.2 Deep learning-based classification framework

Pertaining to the evaluation of the proposed multi-input CNN model, we used the dataset in Section 4.3.2. The training set of images was sampled into a set of samples by using the superpixel labelled images; after computing the superpixel, the corresponding centroid of each superpixel is obtained. We extracted a $3 \times 3 \times 255$ ($W_S = 3$ and $N_{bands} = 225$, as in Section 4.3.5) 3D window around the centroid of each superpixel in the training images. The first and last five bands were removed, as described in Section 4.3.1, after reflectance calibration (the input range is between 0 and 1). In fact, the superpixels are only used to prepare the training samples; in testing, the same window size is selected around each pixel to predict the class of this pixel, which means that no preprocessing needs to be applied in the test phase.

In Section 4.3.5, the proposed multi-input CNN model has two inputs. For the first, we computed the mean spectrum of the selected windows and then reshaped it into (225×1). The second input is prepared as $(3 \times 3 \times 6)$ by masking the whole 3D window by the selected optimal bands.

In the proposed multi-input CNN model, there are four important tunable parameters: the number of feature maps, kernel sizes, pooling and striding sizes and the size of the spatial window. For selecting the optimal values of these parameters, we use an experimentation approach by comparing the results of the proposed CNN model with the results of the SVM model in our testing dataset. Thus, the selected parameters will be optimal for our application and our private dataset.

Firstly, to find the optimal parameters of network head 1, we fixed the parameters of network head 2 and the fully-connected layer. The network head 1 has the following specific tunable parameters: (1) Kernel size with five candidate values $\{3, 5, 7, 14, 24\}$. (2) Number of feature maps with five candidate values $\{5, 10, 15, 20, 25\}$. Stride size with two candidate values $\{1, 2\}$. It should be noted that the pooling size in the proposed CNN model was assumed to be fixed and equal to 2. To select the best values from these candidates, we fine-tuned one parameter and fixed the rest. The best value of the parameter was selected by using the testing dataset, where the value that achieves the highest overall accuracy was selected as the best value of the investigated tunable parameter.

A similar approach was used for selecting the optimal parameters of network head 2, where the parameters of network head 1 were set to the selected optimal values. The tunable parameters of network head 2 is defined as follows: (1) Kernel size with five candidate values $\{1, 3, 5, 7\}$. (2) Number of feature maps with five candidate values $\{5, 10, 15, 20, 25\}$. (3) The spatial window size with five candidate values $\{3, 5, 7, 9, 11\}$. (3) Stride size with two candidate values $\{1, 2\}$.

Secondly, after selecting the optimal parameters of network heads 1 and 2, the

numbers of nodes in the fully-connected layer is considered as the global parameter to be tuned by comparing the performance of the proposed model with the performance of the SVM models. This parameter was selected from eight candidate values $\{16, 32, 48, 64, 80, 94, 112, 128\}$. Thus, the final architecture of the whole multi-input CNN model was selected based on an experimental methodology by comparing its performance on the testing dataset against the reference SVM model. Table 4.3 shows the selected optimal values of these parameters and summarizes the specifications of the proposed multi-input CNN model, which also shows the specifications of each hidden layer, including the kernel size in the convolution and pooling operations, and the number of the extracted features from each convolution layer. The proposed CNN model is a lightweight model where the total number of trainable parameters is $145,322$. Thus, the network is easy to train and converge; this increases the probability of generalization on unseen samples.

Table 4.3: The specifications of the architecture of the proposed multi-input CNN model for detection of the adulteration in red meat products.

Layer		**Kernel size**	**Padding**	**Stride**	**Outputs size**	**Activ.**
Input 1 (225×1)	1D-Conv	24×1	No	(1,1)	10@(202×1)	ReLU
	Max pooling	–	No	(2,1)	10@(101×1)	–
	1D-Conv	5×1	No	(2,1)	15@(49×1)	ReLU
Flatten 1	–	–	–	–	735	–
Input 2	3D-Conv	$3 \times 3 \times 3$	Yes	(1,1,1)	5@($3 \times 3 \times 6$)	ReLU
($3 \times 3 \times 6$)	3D-Conv	$3 \times 3 \times 1$	Yes	(1,1,1)	15@($3 \times 3 \times 6$)	ReLU
Flatten 2	–	–	–	–	810	–
Concatenation	–	–	–	–	1545	–
Fully Connected	–	–	–	–	96	ReLU
Output	–	–	–	–	3	Softmax
The total number of trainable parameters is 150,552.						

For training the model, we used mini-batch SGD with the back-propagation algorithm for minimizing the loss function in Eq. (4.17) and updating the weights of each layer. The learning rate of SGD is set as 0.003. As we used a small learning rate and the training processes are repeated for 500 iterations (i.e., 500 epochs), we decided to add a momentum term into the weights updating criteria for accelerating the training processes, where the momentum value was set to 0.90. Figure 4.7 shows the loss of training and validation samples during the training iterations. The learn-

ing curve shows that the validation error converged after 300 iterations, while the training is still decreasing, which means that the model learns generalized features and no overfitting accrues. Both of the losses are converged after 450 iterations.

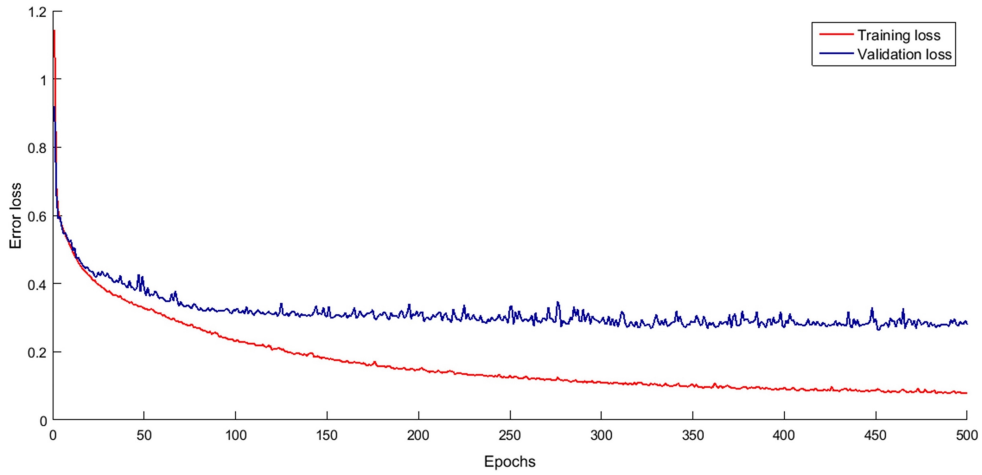


Figure 4.7: The learning curve of the proposed multi-input CNN model.

The performance of the proposed CNN model was evaluated by using the test images. We used the F_1 and overall accuracy to obtain the prediction stability of each class for overall classification. Table 4.4 shows the results of the proposed CNN model. The model achieves good accuracy independently of the state of the meat (i.e., fresh, frozen, thawed, and packed).

Table 4.4: The performance of the proposed multi-input CNN model at each meat status on the test set of images.

Meat status	**F_1 score**			**O.A**
	LAMB	OTHER	FAT	
FSUP	91.2	96.1	98.1	95.4
FSP	82.6	90.1	98.0	90.2
FRUP	92.3	96.1	99.1	95.9
FRP	89.7	94.8	98.4	94.5
THUP	93.2	96.2	98.3	96.1

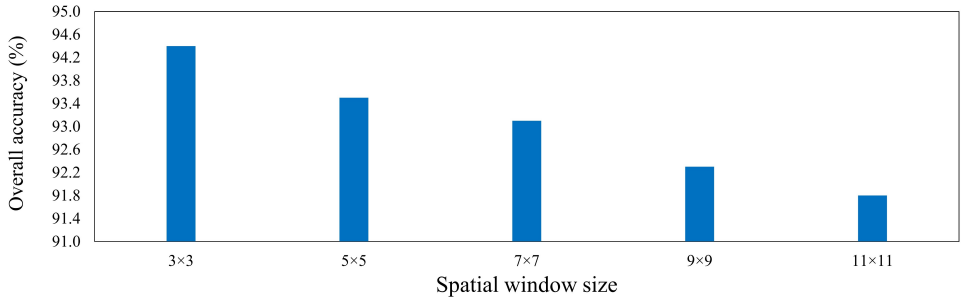


Figure 4.8: The effect of spatial size on the accuracy of the proposed multi-input CNN model.

The proposed CNN model extracts a set of features crossing the spectral and spatial domains. In our experiments, we also investigated the influence of the spatial window size on the whole architecture (i.e., input 2 as in Table 4.3). The results again showed that the best spatial size is 3×3 in applying the model to our application (meat processing). Figure 4.8 shows the influence of the spatial size on the overall accuracy.

4.5 Analysis and discussion

In this chapter, we investigated several methods for detecting adulteration in red-meat products by taking into account the interaction between the chemical composition and textural structure of red meat. Two types of handcrafted features were tested and evaluated by using the SVM model: spectral-based features and joint spectral and spatial features (textural properties of the meat surface). The results in Table 4.1 show that the SVM model outperforms the PLS–DA model in all cases (i.e., the considered feature vectors), Further, Table 4.1 clearly shows that the normalization methods are better than the derivative methods.

The results of the SVM model by using spectral features (as shown in Table 4.1) showed that normalizing the raw spectral by SNV transformation performed better than L_2 normalization and raw spectral on average of all meat states. However, it achieved poor results in the case of frozen-packed and thawed meat (84.1% and 90.7%, respectively as shown in A.1) compared with the other meat states. These unbiased results suggest that the SVM model is still not invariant to the state of the tested red meat.

Adding the textural features into the spectral features slightly improved the accuracy of the SVM model. Table 4.2 shows the performance of the model by adding the texture features into three types of spectral features, individually. The combination of SNV and texture outperformed the other combination (i.e., L_2–norm and raw spectral). Combining the SNV normalization with the texture significantly enhanced the accuracy for FSP and THUP meat statuses, although there was no improvement in the accuracy of the other statuses; for example, in the case of THUP, spectral with SNV normalization achieved 90.7% overall accuracy while both spectral and texture achieved 95.5%. Also, the results of taking the texture properties into consideration showed more reliability in terms of averaging the accuracy over all meat statuses compared with the model with only spectral features.

The proposed deep CNN model achieved excellent results on the test samples. Table 4.4 shows the achieved results at each meat status. The best achieved accuracy was 96.1% when the meat was thawed. The accuracy of the CNN model provided better overall accuracy compared to the other investigated SVM models. Moreover, it provided more stability in the prediction of each class, where the F_1 score increased for all classes. Also, the F_1 score of the OTHER class significantly increased overall; which means that the CNN model is more suitable for detecting the undesired meat (i.e., beef or pork), where this is the main objective of this research.

Table 4.5 provides a comprehensive comparison between the investigated models. In Table 4.5 the average overall accuracies of all meat statuses are provided. Clearly, the results show that the CNN model outperforms the other models, at each and every meat status, in terms of overall accuracy and the F_1 score for all classes. For example, SNV-norm-texture has the best results (92.8% overall accuracy), compared with the other feature vectors, when it is used as a feature vector for the SVM model. The proposed CNN model significantly increases the overall accuracy to 94.4%.

Selecting the best of the proposed models depends on the robustness of the model to be invariant in relation to the meat status, the ability to detect the undesired meat (i.e., the adulteration situation) and the simplicity of the model. The results show that the proposed CNN model satisfies these conditions, where it performed better in the average overall accuracy. In addition, we analysed the variation in the accuracies over all statuses; Figure 4.9 shows the standard deviation of the accuracies for all meat states. The CNN model had the lowest standard deviation (i.e., standard deviation equal to 2.1) compared with the other models; for example, SVM with SNV-texture has 2.3 standard deviation. This result means that the performance of the CNN model is more stable compared with the others.

One of the main advantages of deep learning models is the ability to access and visualize the output of the model at each specific layer. We used this property

Table 4.5: Performance evaluation of all proposed models on average of all meat statuses (i.e., summarizing FSUP, FSP, FRUP, FRP, and THUP).

Model	**Features**	**F_1 score**			**F_1 (avg)**	**O.A**
		LAMB	OTHER	FAT		
SVM spectral–based	Raw spectral	80.3	86.7	96.6	87.9	87.3
	L_2–norm	80.2	86.2	98.0	88.1	87.2
	SNV–norm	86.4	91.1	97.6	91.7	91.4
SVM texture–based	Raw–texture	81.4	87.2	98.0	88.9	88.6
	L_2–norm–texture	83.0	89.6	97.9	90.2	89.8
	SNV–norm–texture	87.5	92.9	98.1	92.9	92.8
CNN deep learning–based	1D-CNN and 3D–CNN	89.8	94.7	98.4	94.3	94.4

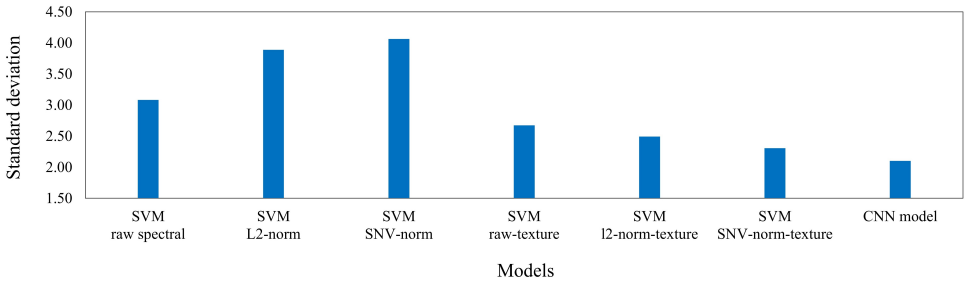


Figure 4.9: The standard deviation of overall accuracies of each model over all investigated meat conditions (i.e., FSUP, FSP, FRUP, FRP, and THUP).

of deep learning models to investigate and visualize the extracted features of red meat by the proposed CNN model. The proposed CNN model, as described in Section 4.3.5, has two heads: 1D-CNN layers for handling the spectral information of meat; and 3D–CNN for processing the textural structure of meat.

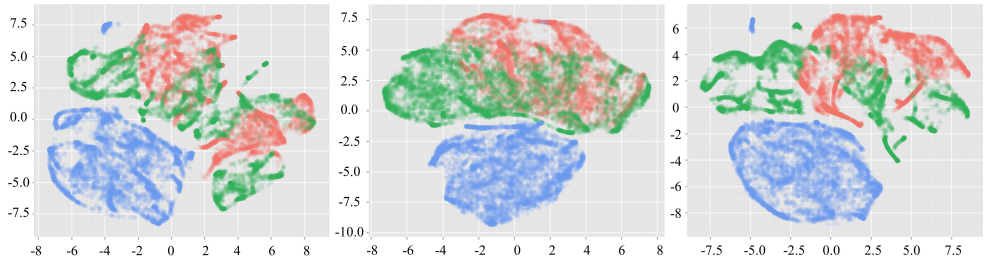


Figure 4.10: T–SNE analysis (scatter plots) of the extracted learned features of the proposed CNN model. *Left*: The features of the first head of the CNN model (spectral features). *Middle*: The features of the second head of the CNN model (textural features). *Right*: The global features (i.e., the shared fully-connected layer). The colours Red, Green and Blue represent classes LAMB, OTHER (beef or pork) and FAT, respectively. The *x-axis* and *y-axis* of scatter plots represent the first and second components of the t–SNE algorithm, respectively.

For visualizing the outcome of these heads, we extracted the output of flatten 1, flatten 2 and fully-connected layers (i.e., layers as shown in Table 4.3), where flatten 1 (735 feature) represents the extracted spectral features, flatten 2 (810 feature) the extracted textural features, and fully-connected (96 feature) the extracted joint features of both spectral and textural.

T–distributed stochastic neighbour embedding (t-SNE) is a robust tool for reducing the dimensionality of the data and visualization [144]. Thus, we adapted t–SNE algorithm to reduce the dimensions of the mentioned feature vectors (i.e., flatten 1, flatten 2 and fully-connected layers) into two t–SNE components. Figure 4.10 shows scatter plots of these components for each set of features.

As shown in Figure 4.10, t–SNE analysis interprets the importance of taking the interaction between the spectral and spatial properties of red-meat into consideration, where spectral features alone (Figure 4.10, left) or spatial features alone (Figure 4.10, middle) are not enough features to discriminate between the classes of this challenging data. Clearly, as shown in Figure 4.10, right, the features of the fully connected layer (the shared layer) are better in terms of class discrimination and representative features of red meat.

Visual comparisons between the models are very important as a kind of qualitative analysis. To generate a classification map by a model, we use a pixel-wise prediction approach by classifying each pixel in the image individually. Figure 4.11 provides the classification maps of the best two models (i.e., the CNN model and SVM with SNV-texture), while all classification maps of all SVM models are available in Appendix A as Figure A.1.

The visual results in Figure 4.11 show the robustness of the proposed CNN model. In Figure 4.11, visually, we see that the detection of the undesired muscles is significantly improved; the visual comparison between SVM and CNN shows that the detection of OTHER class is more accurate and the edges between the meats types are well maintained.

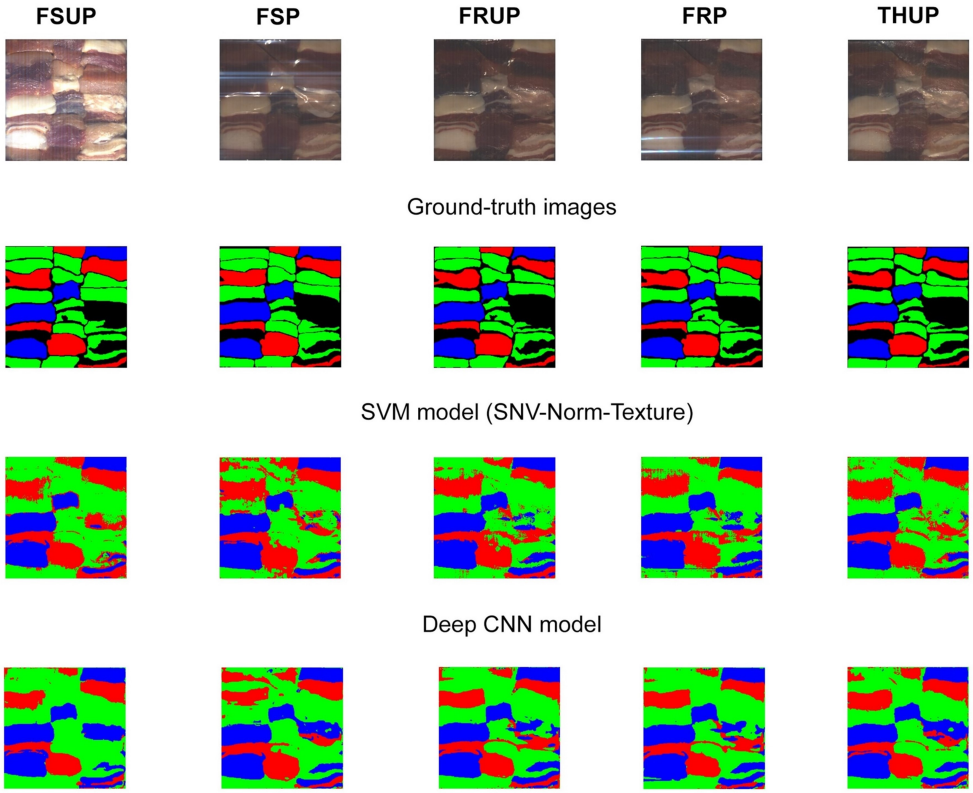


Figure 4.11: Visualization results (classification maps) of the SVM and CNN models provide a visual comparison between the models that have the highest accuracy (i.e., SVM with combining the texture and SNV norm, and the deep learning model with combining 1D and 3D CNN networks). The colours Red, Green and Blue represent classes LAMB, OTHER (beef or pork) and FAT, respectively.

Also, it is worthwhile mentioning that the proposed CNN model is complex in terms of implementation and training phases, where the model has many tunable parameters and needs quite number of experiments to find the optimal architecture for a dataset. However, the proposed CNN model shows a simplicity in the test-

ing phase compared with the traditional machine learning models such as the SVM model. This simplicity could be defined as the ability of the CNN model to handle raw data of images and extract intelligent features from the raw input image. For example, the SVM model used a handcrafted feature: a strong preprocessing (i.e., normalization) and manual feature extraction (i.e., superpixel, GLCM and texture features). The CNN model, however, did not need any preprocessing and presegmentation because the superpixels were used only in the training stage.

Thus, the simplicity of the CNN model and its performance in the testing dataset suggest the model for any real applications in the meat industry, and the fitted model can be applied instantly to new HSI images. However, more research effort is needed to adapt the models for real-time processing, where the used pixel-wise classification approach is time consuming. To classify an image by the pixel-wise approach, the processes are the same, independent and repeated for each pixel. Thus, parallelization of these processes could be considered as a promising research room to enhance the model and achieving real-time requirements of the meat industry.

4.6 Summary

The research, presented in this chapter investigated three types of features for HSI: (1) The chemical composition of materials (presented as spectral information). (2) The textural features of the material surface (presented as the spatial distribution). (3) The interaction between the spectral and spatial features. These types of features were tested and evaluated on the problem of adulteration of red-meat products, yet the same approaches and models can be suitable for other kinds of materials. The overall results show that the spatial information is very important and significantly enhances the HSI classification models; the quantitative and qualitative results, presented in this chapter, demonstrated this fact.

Adulteration of red-meat products is an increasing concern to the industry. In this research, we investigated the robustness of HSI systems for detecting adulteration independently of the state of the products (fresh, packed, frozen or thawed). Different types of spectral and spatial features were investigated by using PLS–DA model-based, SVM model-based and deep learning-based approaches. The deep learning approach included a combination of several layers of 1D CNN, 3D CNN and fully-connected dense layer.

The quantitative analysis shows that the proposed CNN model outperformed the state-of-the-art models (i.e., PLS–DA and SVM models) by achieving an average overall accuracy of 94.4%, and a high and balanced F-score for all classes at all meat statuses. Furthermore, the results show that the CNN model is stable and fairly

invariant to the meat status. Moreover, the CNN is simpler than the SVM in terms of extracting features and testing time. The CNN model is able to handle raw input images without any preprocessing or presegmentation methods.

This chapter shows that HSI systems can be used as powerful tools for rapid, reliable and non-destructive detection of adulteration in red-meat products. Also, this study confirmed that deep learning approaches such as CNN networks provide robust features for classifying the hyperspectral data of meat products; this opens the door for more research in the area of practical applications (i.e., in meat processing or any food processing application).

4.7 Links

This chapter provided a comprehensive evaluation of spectral and textural features of HSI images. Many methods and models were compared and analysed regarding the same research problem (i.e., adulteration in red-meat products) in food processing research domain. The results demonstrated that taking textural features into account significantly enhances the accuracy of classification models.

A deep learning approach using CNN layers achieved the best accuracy, detected by a comprehensive comparison between it and state-of-the-art models (i.e., PLS–DA and SVM models). So, the proposed deep learning model is a robust tool for handling spectral and textural features of meat in a single model. However, the proposed model still processes the HSI data individually (by using the 1D CNN layers for spectral data and 3D CNN layers for spatial information) and only uses a single layer (i.e., the fully connected layer) for joining spectral and textural information of the input HSI data. Moreover, the 3D CNN layers depend on a predesigning step for selecting the wavelengths to represent the textural information, which decreases the ability of the model to extract self and learned features.

In the next chapter, the above research observation will be considered in further detail. Thus, it will focus on investigating the deep learning approaches for fully joining the spectral and spatial (textural) features of HSI images. These fully joint features will be invariant to different HSI systems (i.e., line scanning and snapshot HSI systems). Moreover, the robustness of snapshot HSI systems will be investigated and evaluated against the common line-scanning HSI systems.

Chapter 5

Joint Spectral-Spatial Features for Materials

This chapter proposes a 3D–CNN network for extracting joined spectral-spatial features of materials in snapshot and line scanning HSI data. We also propose a novel graph-based post-processing method for enhancing the prediction of the 3D–CNN approach. As a case study of material classification, we refer to the red-meat classification problem for evaluating the proposed models and methods. Moreover, this chapter provides a comprehensive comparison and benchmark results of three hyperspectral imaging (HSI) systems, line scanning, NIR snapshot and visible (VIS) snapshot HSI, on the same research problem.

Results show that the 3D–CNN model significantly enhances the overall accuracy of state-of-the-art models. Despite the limitations for spectral-only analysis of snapshot HSI, the 3D–CNN model shows robustness in classifying HSI images with an overall accuracy of 96.9% and 97.1% for NIR and VIS snapshot HSI, respectively.

A comprehensive comparison between the three HSI systems shows that the state-of-the-art models provide a competitive accuracy on line-scanning HSI data, while on snapshot HSI data these models are insufficient for achieving accurate classification. The proposed 3D–CNN model (i.e., with the proposed postprocessing method) showed accurate classification for all classes, with average F1 scores of 98.2%, 96.7% and 96.7% for line scanning, NIR and VIS, respectively.

This chapter is organized as follows. Section 5.1 provides an introduction, which includes the motivations and objectives of the research presented in this chapter. Section 5.2 reviews the state-of-the-art techniques which are related to this research. Section 4.3 describes the used dataset, HSI imaging system, HSI image segmentation and the developed methods and models. Section 5.4 provides the experimental setup and results of the proposed framework. Result analysis and discussions are given in Section 5.5. Section 5.6 summarizes the chapter.

5.1 Introduction

Material discrimination using imaging systems is a fundamental and essential task in many computer vision applications such as scene understanding for driving as-

sistance [145,147,148], environmental monitoring [30,146,149] and food processing and grading [19,32,48,55,150,151]. In these applications, the challenge is a discrimination between materials that share similar visual properties.

Each material has its physical, chemical and sensory attributes. Physical attributes are defined by the shape of materials, chemical attributes are the chemical composition of materials, and the sensory attributes are given by the visual and sensing features of materials such as colour, texture or tenderness. These attributes are commonly used for solving fine-grain or general material classification tasks. *Fine-grain material classification* aims to classify the materials that share some or many of these attributes, while *general material classification* is the task for classifying materials that are entirely different with respect to these attributes.

Digital imaging systems (e.g., for recording RGB images) are used for general material classification, but they fail to solve complicated problems such as classifying materials having the same colour and shape. Recently, HSI systems have been introduced to overcome the limitations of colour images. Moreover, HSI systems can combine existing computer vision technologies, designed for RGB imaging, with a *chemometric analysis* for many tasks such as food safety, quality grading or material classification.

In fact, the robustness of the HSI system is possible by providing unique spectral signatures (representation of the chemical composition of material) for each material shown in the image, and also by providing spatial attributes for these materials, attributes like graded quality distribution, the shape of objects, texture properties or the ability of object localization.

So far, the standard way of collecting spectral information in HSI systems was line scanning. In line scanning HSI, the sample moves (on a conveyor belt) and, at the same time, the camera detector detects the reflected light of a particular row (or line) on the sample surface. In this way, the whole sample is scanned and then reconstructed in the collected data as one HSI image. Recently, new hyperspectral sensors [49,50], called *snapshot hyperspectral* cameras, have been introduced as a solution for the limitations (i.e., the hyper-cube acquisition time and the way of generation) in line-scanning HSI systems.

Snapshot HSI systems are designed with the advantages of: (1) Collecting spectral data (i.e., HSI images) at video rate, which make these sensors more applicable for real-time applications. (2) The ability to be a completely portable and movable device (mobile HSI system). However, these new sensors have the drawback of a limitation in the spectral information; this presents a challenge for the application of these modern HSI systems; in Chapter 2, more theoretical and technical information is available (Sections 2.4.1 and 2.4.2 provide details about line scanning HSI and snapshot HSI systems, respectively).

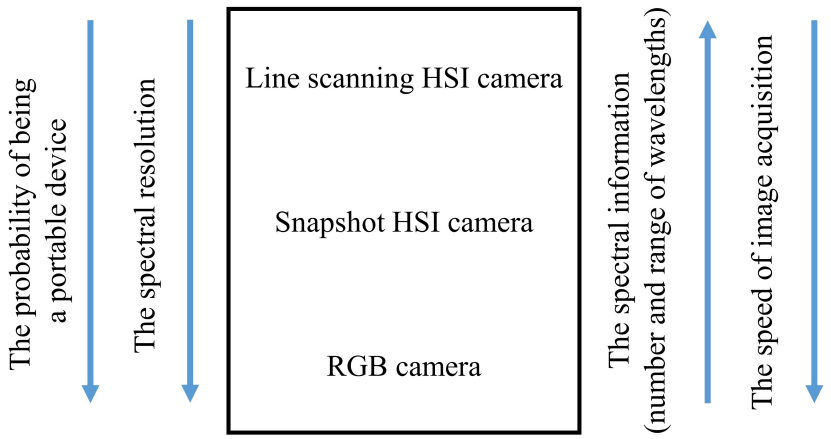


Figure 5.1: A general comparison between line scanning HSI, snapshot HSI, and RGB imaging systems. The arrows point to the increase in the quantity of the mentioned property of the imaging system.

Figure 5.1 shows comparisons between these three imaging systems (i.e., digital RGB, snapshot HSI and line scanning HSI systems) regarding speed, available spectral information, spectral resolution and the way of collecting the spectral information (i.e., the portability of device). Here, it should be mentioned that spectral resolution means the difference between any contiguous wavelengths in the collected spectra and measured in *nm*. In fact, all of these properties depend on the design of the camera sensor and on the applications of the imaging systems.

In case of digital RGB cameras, the probability of being a portable device is the highest as the camera detects the spectral responses of only three regions in the visible region; these responses are then treated as the RGB channels. Thus, the sensor size of RGB cameras is small and only depends on the desired spatial resolution, where the collected spectral information by the RGB system is the lowest in comparison with the other imaging systems. This means that RGB sensors are able to cover large spatial resolution with only an approximation of the three wavelengths from the electromagnetic spectrum as spectral information. Thus, the speed of image acquisition is very high (the highest), and all spectral and spatial information can be collected in a single shot (the highest probability of being a portable device).

Line-scanning systems are designed for detecting more spectral information which can be used for representation of the chemical composition of materials. So, the amount of spectral information (large number of wavelengths and wide range of wavelengths) is the highest in comparison with the other imaging systems. Also, the spectral resolution is the lowest for collecting accurate spectra within a selected

range in the electromagnetic spectrum. The sensor size of the line-scanning camera depends on the amount of spectral information within a one-spatial line. Thus, the speed of image acquisition and the portability of the device are the lowest in comparison with the other systems, where collecting an image needs a special setting such as moving the sample at a fixed speed and fixing the sensor position.

Snapshot HSI systems are proposed to be an intermediate technology between line-scanning HSI and conventional RGB imaging. In snapshot HSI, the amount of spectral and spatial resolution was adjusted and reduced for collecting an HSI image with acceptable spectral information. Thus, the sensor size of snapshot systems depends on both spectral information and the desired spatial resolution. So, a high speed of image acquisition and portable devices are possible and can be implemented with a limitation in the spectral information and the spatial resolution. Thus, a limited amount of spectral information of materials can be collected at a stationary point within a high speed.

For a case study of material classification, we consider red-meat discrimination problems as an example for the proposed framework. Given that red meat (e.g., lamb, beef or pork) share many of the material attributes, this exemplar research is an appropriate case study to evaluate the robustness of classification models. The proposed framework is applicable for solving similar problems for any kind of food products or materials.

Meat processing, regarding quality and safety, is gaining more attention in the meat industry and research. Customers pay more attention to the authenticity and safety of meat as they expect a high-quality product compared with the paid price. Meat authenticity is considered to be one of the safety attributes of meat, where accurate labelling of meat products is important from a customer point of view due to fair-trading or religious reasons. Detection of meat fraud (e.g., mislabelled products) is a challenging task in meat processing plants [33,34]. Practically, red-meat identification processes are usually performed manually (laboratory-based) in the industry, which is time-consuming, and subject to human error. Thus, implementing a rapid, non-destructive, portable imaging system for accurate red-meat classification is very important from both industry and consumers viewpoints.

This chapter aims to investigate the robustness of the new snapshot HSI systems for solving the red-meat classification problem as a case study of fine-grain material classification. Thus, the specific objectives of this chapter are as follows:

- We develop a methodology for data acquisition and sampling of each image into a number of representative samples for modelling.
- We develop and evaluate a deep learning framework for classifying HSI im age data, and for visualizing the robustness of deep learning models against

the state-of-the-art models on three different HSI systems.

- We develop and evaluate a novel postprocessing method to enhance the prediction of deep learning models by using the prediction probabilities and the relationships of contiguous superpixels.
- We provide a comprehensive analysis and comparison, by quantitative and qualitative results, between three hyperspectral imaging systems: Line-scanning HSI, NIR snapshot HSI, and VIS snapshot HSI for the same research problem (i.e., the red-meat classification problem).

5.2 Related work

Material classification models aim to predict a class label for each pixel in the image. Such a model analyses the pixel value, or a combination of pixel values, and some spatial features (such as texture features) of an area surrounding a target pixel. The robustness of colour images for material classification was investigated in [152]. The results showed that a single RGB image is insufficient to discriminate the considered materials. Thus, a bidirectional reflectance distribution model, by collecting many images of an object under different lighting conditions, was used to provide more information about each pixel in the image. These results demonstrate that more information about pixels typically increases the probability of correctly classifying a large diversity of materials based on imaging systems [152].

HSI imaging systems are a valuable tool for providing the chemical components of materials by means of images, where each pixel is a vector of multiple reflectance values (spectral signature) representing the chemical composition of materials in the image. Line-scanning HSI is commonly used for classifying materials (especially in indoor applications) in food processing such as for fine-grain meat classification [77, 79], in agriculture for wheat and kernel classification [150, 151] and in general material classification and sorting [153]. In all of these studies, only spectral features of materials were used (by averaging all pixels in an ROI) as input for classification models like PLS–DA, LDA, PCA or an SVM. Alternatively, textural and spectral features of line-scanning HSI were investigated in [11] for meat processing applications.

The main advantage of the line-scanning method is that it provides in-depth spectral features (typically covering hundreds of bands with a very fine spectral resolution). However, image acquisition is very slow, and a large size of a generated hypercube is computationally expensive [19, 45]. As a solution, snapshot HSI was

introduced; recently it has gained attention in many research areas such as terrain classification for autonomous driving [147, 148], vegetation classification [149] and for object recognition [154].

In [147], two snapshot HSI sensors were investigated for terrain classification with dynamic data acquisition (i.e., moving sensors). These sensors cover a limited number of bands, from VIS to NIR regions of the electromagnetic spectrum. In [148], Gabor texture features were adapted for classifying visible snapshot image data; these features were added to the visible spectral features (i.e., based on pixel values), and then a random-forest algorithm was used for evaluating the features. In both [147, 148], experimental results showed that snapshot HSI sensors are valuable and competitive tools for handling pixel-wise classification problems such as terrain classification for autonomous driving systems.

Recently, deep learning approaches for supervised learning are considered state of the art in many computer vision applications dealing with detection, tracking, recognition or classification. Deep CNN models show robustness as a feature extractor from raw input data. Deep CNN models are applied to HSI images for object recognition [154] and remote sensing [129, 130, 139]. Moreover, CNNs show flexibility in dealing with HSI data by introducing a 1D-CNN [129] (designed for processing spectral inputs), 2D–CNN for single wavelength images or PCA-component images [130] and 3D–CNN for an intelligent combination of spectral and spatial image data [130, 134, 139].

In [130, 139], the same approach was used by stacking several 3D–CNN layers followed by a set of fully-connected layers; the difference between these architectures is the size of the spatial domain in the input layer, the kernel representation and the size of the whole model (i.e., the total number of trainable parameters). The proposed 3D–CNN architecture [139] showed that small input size and lightweight models are more accurate and efficient for classifying HSI image data. In [134], a slightly different 3D–CNN model was proposed, where the padding and striding operations were used for downsampling the spatial domain into one unit, then 1D-CNN layers were used.

5.3 Materials and methods

5.3.1 Dataset and sample preparation

We used a collection of fresh red-meat samples from different local supermarkets. The total number of samples is 184, including lamb (67), beef (73) and pork (44). All of the samples were chosen from loin and leg chops. The samples were chosen from eight different commercial products which are commonly found in local shops.

Table 5.1 shows the considered product types and provides details about the number of samples for each product. Figure 5.2 shows some samples from the considered commercial products.

Table 5.1: Number and product type of the collected samples of each meat type.

Meat type	Product type	Number of samples
Lamb	Loin chops	46
	Leg chops	21
Beef	Minute steaks	36
	Eye fillet steaks	10
	Porterhouse steaks	18
	Scotch fillet steaks	9
Pork	Loin chops	26
	Leg steaks	18
The total number of samples is 184		

The selected set of samples includes more than 13 muscle types based on meat standards [155]. Figure 5.3 shows the lamb muscles that are considered in this research and reveals the distribution of muscles in both the loin and leg chops of a lamb carcass. In [155], more descriptions were provided about muscles types and structures in both loin and leg chops. Then, the collected samples were randomly separated into training and testing sets as follows: A set of 105 samples was used for training and processing; remaining samples were used only for testing and evaluating the proposed methods and models.

5.3.2 Hyperspectral imaging system

Two HSI systems were implemented for imaging the meat samples, snapshot HSI and line-scanning HSI. The meat samples were prepared for imaging by simply drying the meat surface with normal tissues. Then, the whole chops were scanned by using line-scanning HSI first and then by using snapshot HSI. The meat samples were not cut into specific types of muscles or specially prepared for imaging as in [77, 79, 82]. Thus, we keep the same shape and texture of chops as in the supermarkets, which makes the collected HSI images more reliable and applicable for real applications.

Figure 5.2: Selected colour images of red-meat samples from the collected samples dataset.

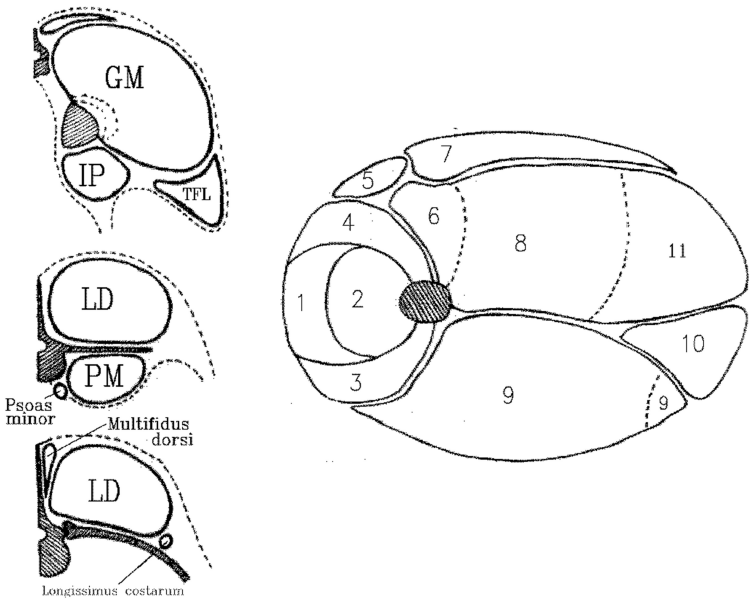


Figure 5.3: Muscles distribution in lamb meat chops. Left: The distribution of muscles in loin chops - longissimus dorsi (LD), psoas major (PM) and gluteus medius (GM). Right: The distribution of muscles in leg chops, where 1 = rectus femoris, 2 = vastus intermedius, 3 = vastus lateralis, 4 = vastus medialis, 5 = sratorius, 6 = pectineus, 7 = gracilis, 8 = adductor, 9 = biceps femoris, 10 = semitendinosus and 11 = semimembranosus. Images were copied and adapted from [155].

The line-scanning HSI system, as illustrated and described in Section 2.4.1, was used to collect a set of 81 line-scanning HSI images with two different meat samples recorded in the same image on average, composed of 51 and 30 HSI images of training and test meat samples, respectively.

The snapshot HSI system, as illustrated and described in Section 2.4.2, was developed by using the two snapshot HSI sensors: NIR and VIS. For imaging, each sample was placed on the conveyor belt while the snapshot HSI cameras collected a sequence of images of each sample ensuring that all sample portions are covered in these sequences; Figure 5.4 shows an example of these sequences that was used for covering the portions of two lamb loin samples. The average number of HSI images (i.e., a sequence of images) per sample was six. It should be noted that the samples could have been scanned at a stationary position; however the conveyor belt was used to simulate a practical situation in a meat processing plant.

In fact, using the sequences and the conveyor belt is for the following purposes: (1) To collect a set of representative images covering all portion samples as the field of view of the system is smaller than the size of the samples. (2) To obtain representative HSI data under different illumination conditions. For example, the data of the same meat portion was collected at a different location in the field of view of the system, which means that the collected data covers more information about the distribution of light in the field of view of the system. (3) To increase the size of the dataset (i.e., the number of images).

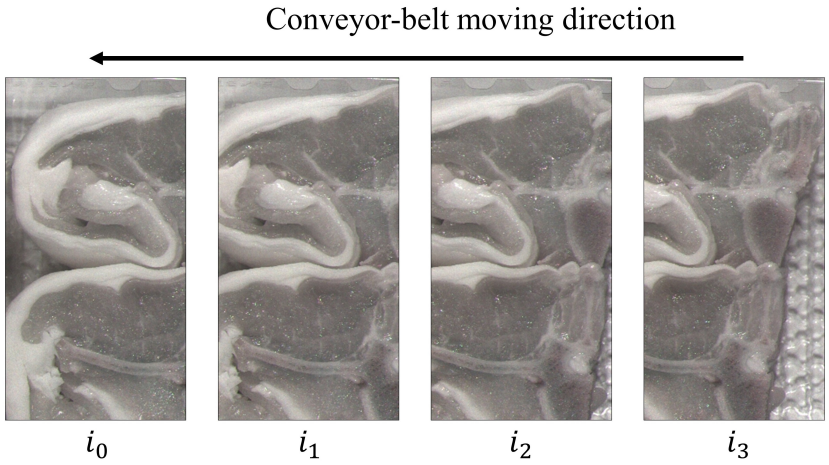


Figure 5.4: Example of snapshot HSI images (false-colour images) acquired for representing two lamb loin-chop samples. The sequence in the figure represents the time of acquisition during the motion of a conveyor belt.

The total number of images in the collected set of snapshot HSI images is $1,800$, including 600 images obtained from the training meat samples for the NIR and VIS cameras and 300 images from the testing meat samples for the NIR and VIS cameras. This approach, and the number of images, defines a consistent dataset for deep learning-based models.

The line-scanning and snapshot HSI systems for acquiring all HSI images used in this chapter are described and demonstrated in more detail in Chapter 2; the system setup, parameters and reflectance calibration are explained in detail in detail for line-scanning and snapshot HSI, respectively (Sections 2.4.1 and 2.4.2).

5.3.3 HSI segmentation and processing

The main task in this work is developing a pixel-based classification model for red-meat products. The approach of considering just the mean spectrum of each meat type for building such a classification model, as proposed in [77, 79, 82], is inefficient for the following reasons: (1) Considering different muscles (such as 13 in this research), by averaging we lose the spectral patterns. (2) Averaging the spectral response of meat for a whole sample does not represent the variations of light and meat across the sample and the texture of the meat samples. (3) This approach is inefficient for complex learning models like deep learning due to a limitation in the number of samples.

Thus, we develop a methodology for re-sampling HSI images into a set of representative *data items* (e.g., regions from each muscle shown in an image). These selected data items are then used to extract a set of features. We also use the SLIC algorithm [101], as described in Section 4.3.4, for segmenting HSI images into a set of labelled segments, where one segment (i.e., one superpixel) consists of a group of connected pixels that share some spatial and spectral properties.

For line-scanning images, the PCA space was used for computing the similarity between pixels while generating the superpixels. For snapshot images, the whole spectral vector was used in the SLIC algorithm. Figure 5.5 (c) shows an example of SLIC superpixels generated from a snapshot HSI image.

For having the GT data, the images were manually labelled into meat and fat classes. Then, the segmentation map and the GT are matched for generating a GT superpixel map. For each superpixel, the centroid of the segment [95] is computed; then, its coordinates were used for deciding whether it is meat, fat or background. Figure 5.5 sketches the proposed methodology. By using this methodology, a set of data items were generated from the training datasets of line-scanning and snapshot images.

In the SLIC algorithm, the superpixel segmentation map is controlled by the size

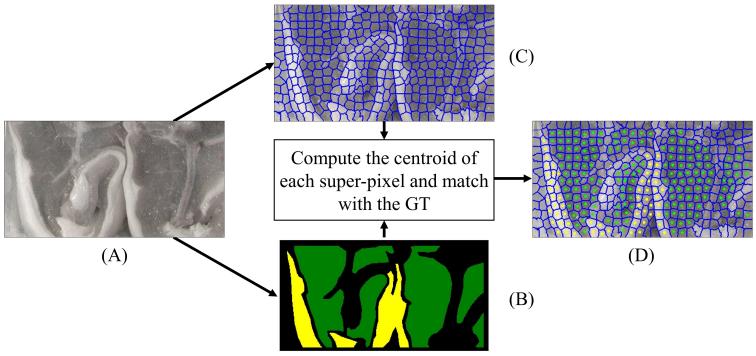


Figure 5.5: Methodology for re-sampling an HSI image into a set of representative points. (a) An HSI image of two lamb samples (an extracted false-colour image of an input hypercube). (b) GT image manually generated by using a multi-polygon tool. (c) Superpixel segments of the input image. (d) Selected points from each class; colours Green and Yellow denote meat and fat, respectively.

and shape of superpixels as control parameters (number of superpixels and compactness factor) [101]. In our approach, these parameters have an important impact, where the size of superpixels defines the number of selected data items in training images, and the compactness factor controls the positioning of the centroids. We empirically selected 145 and 0.4 for superpixel size (in pixels) and compactness, respectively.

Experimentally, we found that these settings provide a reasonable number of selected data items (covering the variation in the training images) and regular shapes of superpixels, as shown in Figure 5.5 (c). Thus, these settings are fixed for all experiments in this chapter for a fair comparison between models and HSI systems. Table 5.2 shows the numbers of training data items for each class (here, the classes are lamb, beef, pork and fat) in both datasets. These data items were randomly separated into 70% for training purposes and 30% for validating the models.

Table 5.2: Numbers of selected data items (at centroids) for each class, for training images in the line-scanning and snapshot datasets.

HSI system	**Lamb**	**Beef**	**Pork**	**Fat**
Line scanning	3909	6297	3955	1943
NIR snapshot	57241	75325	50030	30730
VIS snapshot	96321	112213	77049	44019

5.3.4 Deep 3D–CNN for HSI classification

Snapshot HSI cameras are designed as on-chip micro-multi-spectral detectors. They usually cover a limited number of wavelengths. This limitation in spectra of snapshot HSI inspired us to investigate complex models for dealing with the challenges in spectral information.

The main objective of this thesis is to investigate methods for addressing the interaction of textural and spectral information of meat that is provided by HSI systems. In Chapter 4, a combination of 1D CNN and 3D CNN was proposed to extract learned features of meat from HSI images. The proposed model used 1D CNN layers to extract the spectral features and 3D-CNN to extract the spatial (or textural) features of meat. The proposed model achieved good results in comparison with traditional approaches for classifying HSI images of meat. However, the proposed model still utilizes the HSI features individually (by using the 1D CNN and 3D CNN layers) and only uses a single layer for joining spectral and textural information of the image data.

This chapter aims to investigate a new and deeper way to extract fully joined features from the textural and spectral information of meat in HSI images. Extracting the joined features in a single operation and model instead of combining multiple operations (i.e., 1D CNN and 3D CNN) could make the model extract robust features and provide a better joining of the HSI image information (i.e., spatial and spectral information). This research observation inspires us to propose a deeper and novel deep learning framework to extract the fully joined features of red-meat types in HSI images.

In this chapter, we propose the 3D–CNN approach for the extraction of useful and joined features of snapshot HSI images. We evaluate these features for solving the red-meat classification problem and compare their results with a reference HSI system; we choose a line-scanning HSI system for reference. Moreover, we evaluate and compare the proposed approach with the existing studies' models for red-meat classification [79]. These models were implemented in this chapter in the same way as in the published research as they are considered to be state-of-the-art models. Also, we implemented an SVM [115] model for comparison purposes because the SVM is a common reference for machine learning.

The main task of the proposed deep learning model is to classify the content of an HSI image by using pixel-based classification approach. Thus, the model uses the local spectral and spatial features of meat (i.e., pixel value and the area surrounding it) for obtaining the relevant class of that pixel (i.e., belonging to lamb, beef, pork or fat). By this approach, the input of the model is defined as a fixed 3D spatial window around the target pixel. In the training phase, the spectral and spatial features were

extracted from the images as well as their ground truth, as described in Section 5.3.3. In the testing phase, the model was directly applied on each pixel (i.e., as a 3D window) individually for obtaining the classification map of an HSI image.

Architecture of the 3D–CNN model

In fact, 3D–CNN operations were originally introduced for handling the spatio-temporal features in video sequences such as in action recognition applications [122]. HSI images represent the spectral features in the form of a 3D image. Thus, 3D–CNN operations for HSI image data are defined by convolving a set of 3D kernels for producing a set of 3D feature maps. At l-th layer, a value of the j-th feature map at location (x, y, z) is computed as follows:

$$v_{lj}^{xyz} = f\left(\sum_{m=0}^{M-1}\sum_{i=0}^{w_l-1}\sum_{n=0}^{h_l-1}\sum_{k=0}^{d_l-1} k_{ljm}^{ink} v_{(l-1)m}^{(x+i)(y+n)(z+k)}\right) \tag{5.1}$$

where v_{lj}^{xyz} is the value at position (x, y, z) of the j-th 3D feature map in the l-th layer of the network; v is a value in a feature map and not related to input layer. Triple (w_l, h_l, d_l) defines the size of the 3D kernel k_{ljm} in the l-th layer (i.e., w, h, d for height, width and depth, respectively), connected to the mth feature map in the previous layer.

The 3D–CNN operations, as defined in Eq. (5.1), are convolution operations between the current and previous layers. Then, an activation function is applied on the output of each layer. In the proposed model, we use ReLU functions for activating the output of 3D–CNN layers. Thus, the function $f(\cdot)$ in Eq. (5.1) is a ReLU function as defined in Eq. (4.15).

Visual geometry research group (VGG) investigated the idea of CNN blocks in the implementation of a deeper CNN model, named VGGNets [173], for object recognition and image classifications. In the VGGNet approach, the proposed model is structured as multiple CNN blocks, where each block consists of two CNN layers followed by one max-pooling layer with a pooling size of 2 and 2 steps striding. The max-pooling layer is used to downscale the features for extracting robust features with different scales with respect to the input data [173].

The proposed approach in VGGNets inspired us to investigate the idea of structuring the 3D–CNN layers in the form of multiple 3D–CNN blocks. Thus, we propose the 3D–CNN blocks to extract fully joined features of meat from HSI image data. For the classification part, a fully connected layer and output layer (softmax function) need to be connected to the output the 3D–CNN blocks. Thus, the number of 3D–CNN blocks is a tunable parameter and needs to be defined. It should be

noted that in the proposed approach we fixed the number of fully connected layers into one layer.

For selecting the number 3D–CNN blocks, we followed a similar approach to that discussed in Chapter 4. Thus, the number of 3D-CNN and fully connected layers were empirically selected based on observing the accuracy of a cross-validation evaluation on training data. In this step, the parameters of all layers (i.e., all layers including CNN, pooling and fully connected) were fixed, assuming that this initial evaluation is only to approximately estimate the best number of 3D–CNN blocks. Based on this initial evaluation, we empirically found that only two 3D–CNN blocks are the optimal values for our dataset.

The proposed 3D–CNN model, as sketched in Figure 5.6, consists of a hierarchical structure of nine layers with the following structure: (1) An input layer is defined as a 3D window surrounding the target pixel in the image. (2) Two 3D–CNN blocks, each block consists of two 3D–CNN layers and one max-pooling layer. (3) A fully-connected dense layer as a feature extractor for red meat. (4) An output layer as a fully-connected softmax layer for having the probability of one of the meat types.

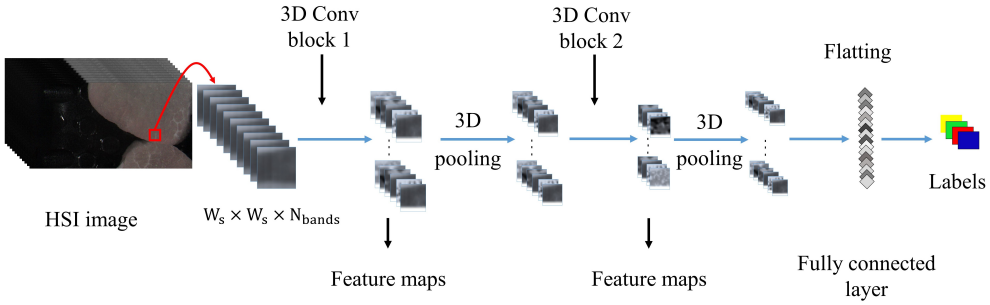


Figure 5.6: Schematic demonstration of the proposed 3D–CNN model showing the main structure of the model. The detailed specifications of the model for each HSI system are provided in Tables 5.3, 5.6, and 5.9

In the proposed architecture, each convolution block has the same kernel representation and the same number of feature maps. Padding (with the same values) and striding (with a stride of 1) operations are used for keeping the same output at each block. For downsampling the feature maps, 3D max-pooling layers are added after each convolution block.

A dropout technique is proposed as a regularization method [156]. The technique aims to approximate the training of a large number of neural networks with different architectures by dropping out some nodes in the network. In the training phase, a number of layer nodes (defined as a ratio of nodes to be dropped out) are temporally removed along with all their incoming and outgoing connections, which makes the layer (or the architecture) look like it is training with a different number of nodes at each update in training. In fact, dropout has an effect on the training process by making it noisy and roughly doubling the number of epochs required to network convergence. However, dropout is useful in making the activations of the hidden units become sparse. It should be noted that it is applied only during the training process, while in the prediction phase, the output of the architecture is computed without dropping out any nodes.

To avoid overfitting in our model, we used the dropout technique [156] by adding three dropout layers after the pooling and the fully-connected layers. Moreover, we sought to reduce the whole model size by tuning a small number of kernels, which is good for generalization [139, 156].

Training procedure for the 3D–CNN model

3D convolutions are able to handle and extract learned features among spectral and spatial domains. Multiple 3D–CNN layers extract robust and complex features in a hierarchical way. Thus, the proposed 3D–CNN model is implemented by the following four main steps:

Step 1: Target samples extraction: The model was implemented for pixel-wise classification. Thus, the input shape is fixed as a window around the target pixel. The data items, as described in Section 5.3.3, were used for extracting a set of 3D windows (around each item) of size $W_s \times W_s \times N_{band}$; as the target sample; W_s denotes the spatial size. In training the proposed 3D-CNN model, the extracted target samples were randomly separated into 70% for training and 30% for assessment purposes and best model selection.

Step 2: Forward propagation. In this step, a sample of size $W_s \times W_s \times N_{band}$ is fed into the first convolution block with kernels of size $K_1^1 \times K_1^2 \times K_1^3$. Using Eq. (5.1), each 3D–CNN layer in block 1 produces M_1 feature maps of size $M_1 \times W_s \times W_s \times N_{band}$. Next, the resulting feature maps are loaded into the first 3D max-pooling layer with a pooling size of $2 \times 2 \times 2$ and strides of (s_1^1, s_1^2, s_1^3), which reduces the size of feature

maps into a size of $M_1 \times W'_s \times W'_s \times N'_{bands}$, where

$$W'_s = W_s/s_1^1 \quad \text{and} \quad N'_{bands} = N_{bands}/s_1^3$$

Next, the maps are fed into the second convolution block with $K_2^1 \times K_2^2 \times K_2^3$ kernels, and by the same way it produces a new set of feature maps with a size of $M_2 \times W'_s \times W'_s \times N'_{bands}$.

The second 3D max-pooling layer, as in the first pooling layer, again reduces the size of maps into $M_2 \times W''_s \times W''_s \times N''_{bands}$, while the pooling size here is also $2 \times 2 \times 2$ and the used strides are (s_2^1, s_2^2, s_2^3). The flatten layer converts the feature volumes into a vector of size $1 \times M_2 \times W''_s \times W''_s \times N''_{bands}$. This set of features is used to compute 128 features of the fully-connected layer.

Subsequently, the output layer of size 1×4 (4 is the number of classes) is computed by using the previous 128 features. The resulting vector is passed into a softmax function for obtaining the probability over the predefined classes as the final output of the model.

Step 3: Back propagation. In the back propagation, the weights of the model were adjusted and updated using a stochastic gradient descent approach after each forward propagation. In these processes, the main task is to minimize a loss function between the actual output (i.e., ground truth) and the model output (i.e., a prediction in the forwarding process). In the proposed model, we use the *adaptive moment estimation* (Adam) optimizer [157] for minimizing the categorical cross-entropy loss function, which is computed as in Eq. (4.16).

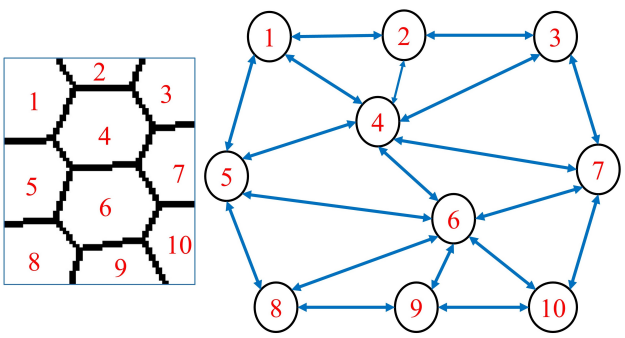
Step 4: Model validation. In training processes, the training dataset is divided into patches, where each patch is a group of samples and the size of the patch is equal to the number of samples in the patch. In each forward and backward process, a batch of samples is selected and fed to the model for computing the predicted output (forward propagation). The mean loss value of these samples is computed by using Eq. (4.16). Then, the weights of the model are updated to minimize the resulting loss (back propagation). These processes are repeated for n times to train the model on all patches, where n is the number of patches. The process of passing all patches for training is called an *epoch* of training. Thus, the epoch means that the model is trained on the entire training dataset only once. At the end of each epoch, the validation samples are fed to the model, then the mean loss of validation samples is computed and used for selecting the best model.

Graph-based postprocessing method

In the proposed 3D–CNN model, the local spectral and spatial features are used for classifying each pixel (*pixel-based* classification) of the image. Also, superpixel segmentation can be used for improving the prediction of the model; for example, classifying each superpixel (*superpixel-based* classification) by taking the 3D window around its centroid instead of classifying each pixel. A superpixel-based classification method has two main advantages: It is computationally efficient by reducing the number of classification times, and more spatial information is taken into consideration. However, it has a high sensitivity to the shape of a segment and the location of the centroid, which reflects badly in some cases on the classification results.

The output of the 3D–CNN model is a probability distribution over the predefined classes. Thus, we propose a novel method for postprocessing the output of the model by taking the relationships and connectivity of the adjacent superpixels into account; the method can be applied to any deep learning model for pixel-wise classification. The proposed method aims to compute a joined probability between the target superpixel and its neighbours.

Thus, the method uses a set of weights for defining the contributions of the target superpixel and its neighbours to enhance the superpixel-based classification method in a systematic way (*weighted superpixel-based* classification). Undirected graphs are used for representing the connectivity between adjacent superpixels. Figure 5.7 shows an example of a superpixel segmentation map, its undirected graph and the adjacency matrix of the graph for defining the connectivity between segments.



	1	2	3	4	5	6	7	8	9	10
1	0	1	0	1	1	0	0	0	0	0
2	1	0	1	1	0	0	0	0	0	0
3	0	1	0	1	0	0	1	0	0	0
4	1	1	1	0	1	1	1	0	0	0
5	1	0	0	1	0	1	0	1	0	0
6	0	0	0	1	1	0	1	1	1	1
7	0	0	1	1	0	1	0	0	0	1
8	0	0	0	0	1	1	0	0	1	0
9	0	0	0	0	0	1	0	1	0	1
10	0	0	0	0	0	1	1	0	1	0

Figure 5.7: Schematic representation of connected superpixels, their undirected graph and the adjacency matrix presenting the graph. *Left*: Example of connected superpixels. *Middle*: Undirected graph representation showing the connectivity between the segments. *Right*: Adjacency matrix presenting the graph.

For demonstrating the method, let H be an HSI image to be classified, SP is a set of superpixel labels of H from 0 to S, and P is the model output of all superpixels. For each element in SP, neighbouring superpixels SN are extracted using the undirected graph and the adjacency matrix of SP, where SN is a sub-set (of N elements) from SP defining the labels and the number SP_i of neighbours of the i-th superpixel. P_i is a vector of C elements consisting of the probability of the i-th superpixel for being one of the predefined classes C, where $\sum_{j=0}^{C-1} P_i(j) = 1$. PN is the model output of SP_i neighbours. Thus, the joined probability P_i' between SP_i and its N neighbours is computed as follows:

$$P_i' = w_t \cdot P_i + w_n \cdot \sum_{k=0}^{N-1} PN_k \quad , \quad \text{with} \tag{5.2}$$

$$w_n = (1 - w_t)/N \tag{5.3}$$

where w_t and w_n are weight factors defining the contribution of the target superpixel and the neighbours for the new joined probability P_i', respectively. The new joined probability P_i' vectors show the probability distribution over the predefined classes, with more consideration about the correlation in the spatial domain, where $\sum_{j=0}^{C-1} P_i'(j) = 1$. The parameter w_t is a tunable value and fixed for all target superpixels. Note that the multiplication in Eq. (5.2) is a scalar by vector operation, and the addition is an element-wise addition of two vectors. Then, the prediction of superpixel SP_i is done by an arg-max function as follows:

$$y_i = \arg\max(P_i') \tag{5.4}$$

where y_i is the predicted label (or class) of the superpixel SP_i.

5.4 Experiments and results

The testing datasets of images (i.e., the HSI images of the 79 testing meat samples by each HSI system), as described in Section 5.3.1, were used for generating the final results of the proposed 3D-CNN model as well as the baseline models. The results represented in this section and the next section were obtained from independent meat samples and were not included in any stage of the model development simulating the real-world application of the study.

For comparing with the state-of-the-art models, we use two baseline models PLS–DA and SVM with RBF kernel as the classification models. In both SVM and PLS–DA, we used 10-fold cross-validation with grid search for hyper-parameters tuning as follows: the penalty parameter and RBF kernel coefficient in SVM training; and the number of components in PLS–DA training. In both PLS–DA and SVM

implementation, we followed a similar implementation (i.e., same experiments) as in existing published studies [77, 79, 82]. Thus, the mean spectrum of meat samples (or region of the samples) were extracted and then used as spectral features for classification by PLS–DA or SVM. All the implementations of these models were done using R-analysis and Python for implementation. The proposed deep learning framework was implemented on KERAS with a TensorFlow backend.

In the proposed 3D–CNN model, there are four important tunable parameters including the number of feature maps, kernel sizes, pooling and striding sizes and the size of the spatial window. For selecting the optimal values of these parameters, we use an experimentation approach by comparing the results of the proposed CNN model with the results of the baseline models in the testing dataset. Thus, the selected parameters will be optimal for our application and our private dataset.

In optimizing these parameters, we used an NIR snapshot dataset to find the optimal values of the parameters of the model, then the same model (i.e., same architecture) was adapted to select the architectures of the line-scanning and VIS snapshot datasets; the NIR snapshot dataset was selected because of the amount of spectral (25 wavelengths) in the NIR snapshot data is intermediate between line-scanning (225 wavelengths) and VIS snapshot data (16 wavelengths).

For selecting the optimal value of one parameter, we fixed the rest of the parameters and fine-tuned the single parameter by using a range of values. Then, the optimized value of that parameter is used in the further fine-tuning processes of the other parameters. The used range of values for each parameter is as follows: (1) Kernel size with four candidate values $\{1, 3, 5, 7\}$ for both height and width of kernels, and five candidate values $\{3, 5, 7, 9, 15\}$ for the depth of kernels. (2) Number of feature maps with five candidate values $\{2, 4, 8, 16, 32\}$. The stride size of the CNN layers was assumed to be equal to one for all dimensions. (3) The pooling size of pooling layers was fixed to be 2 for all dimensions, while the striding size of these layers was optimized from three candidate values $\{1, 2, 3\}$. (4) The number of nodes in the fully connected layer was selected from five candidate values $\{32, 64, 128, 256, 512\}$. (5) The spatial window size with five candidate values $\{3, 5, 7, 9, 11\}$. Thus, the final architecture of the 3D–CNN model was selected based on an experimental methodology by comparing its performance on the testing dataset against the reference baseline models.

For evaluating the proposed framework and the baseline models, we used the standard F_1 score, overall accuracy (O.A), and *average accuracy* (A.A) (defined as average of recalls of all classes) as measures to evaluate the performance of models; all mathematical definitions of these metrics are provided in Section 4.4. For obtaining the evaluation measurements, the test sets of HSI images (from line-scanning and snapshot systems) were handily labelled for generating the GT images; Scyven

(software for HSI images analysis and exploration) was used for generating the GT images. Then, we extracted all labelled pixels from each GT image and the same pixels of the resulting classification map for computing the evaluation measures (*all data evaluation*).

In the case of the 3D–CNN model, we further investigated the performance of the model by evaluating each testing image individually and then averaged the results of all images (*average images evaluation*). We evaluated the model in this way to show the robustness of the model for any real applications by using snapshot HSI images in food industry.

In all experiments, the parameters of the SLIC segmentation algorithm were empirically selected as 145 and 0.4, for superpixel size (in pixels) and compactness factor, respectively. Also, the weight factor w_{tg}, as in Eq. (5.2), was set to be 0.3. For training the 3D–CNN models, we used Adam optimizer [157] with a learning rate of 0.0001 and patch size of 256. The models were trained for $1,000$ epochs.

5.4.1 Spectral signatures visualization

For extracting the spectral signature of each class (i.e., meat types and their fat), the selected data items were used by cropping a $5 \times 5 \times N_{bands}$ window around each item. Then, the mean spectrum of each window was computed for obtaining the spectra of each window. Next, the mean of all spectra of each class was computed for having just four spectra, one for each class.

Figure 5.8, left, shows the spectral signatures of each class in the HSI systems. For quantifying the relationships between signatures, we use the *Pearson correlation coefficients* for showing the similarity between signature pairs, where high positive values mean that a linearity between class pairs is high.

As shown in Figure 5.8, right, line-scanning signature pairs have a low correlation, which means that the probability of having discrimination patterns is high. In the NIR snapshot, the correlation between lamb and pork signatures is very high (approximately 1.0), which means that they have an identical shape.

Thus, the only difference is in the intensities, which decrease the probability of accurate discrimination due to the impact of light scattering on spectra. Similar results are shown for the VIS snapshot data, where the coefficients of lamb-pork and beef-pork are very high. These observations show that the sole use of the spectral features of HSI snapshots is insufficient for achieving a high accuracy for those material-based classification problems.

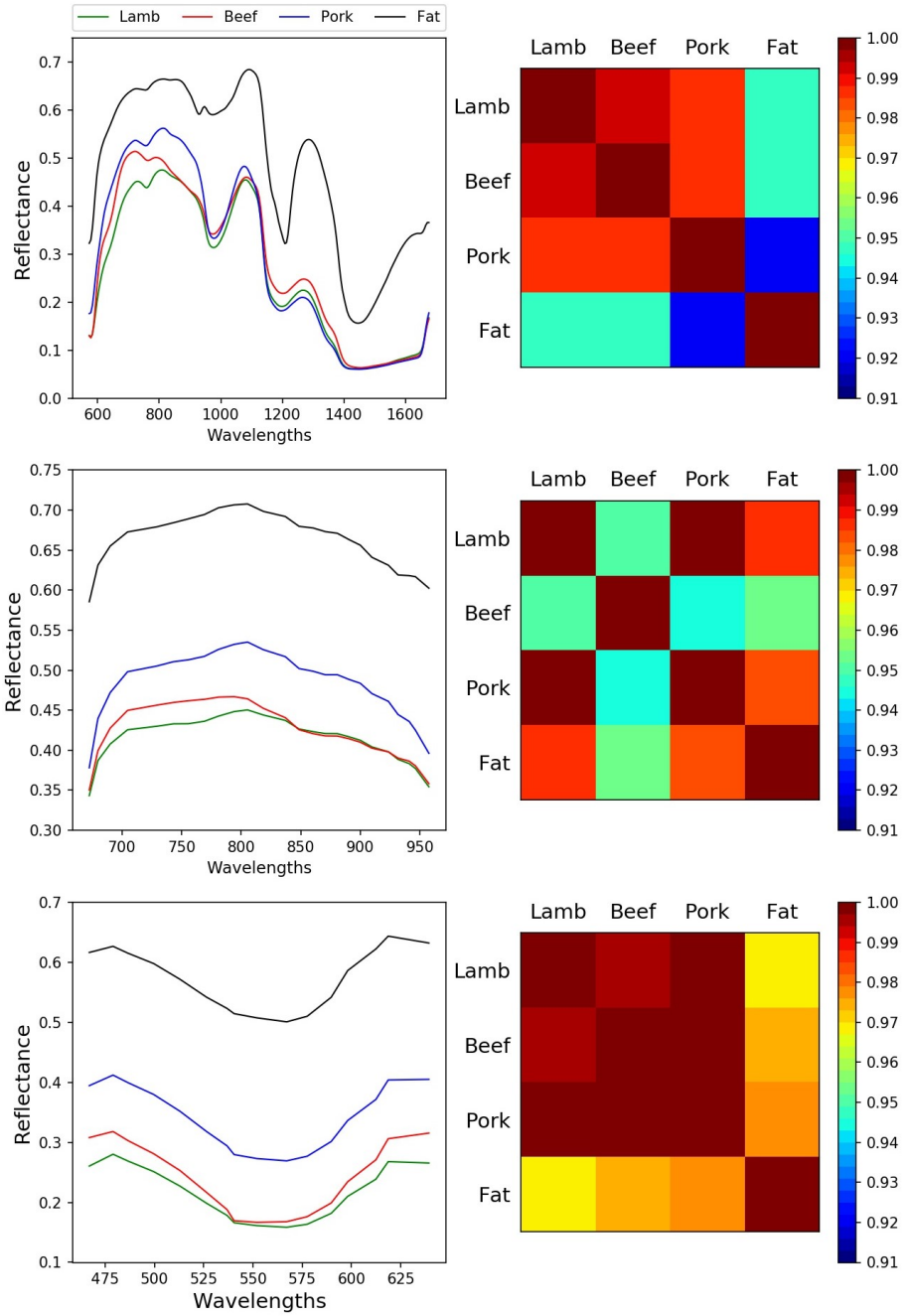


Figure 5.8: Spectral signatures and correlation analysis of red-meat types. *Left*: Extracted spectral signatures of red-meat types and their fat. *Right*: Correlation-coefficient matrices of each HSI system showing the similarity between (or the dependence on) signatures pairs; see colour index on the right for Pearson correlation coefficients. From top to down: Figures present line scanning, NIR snapshot and VIS snapshot, respectively.

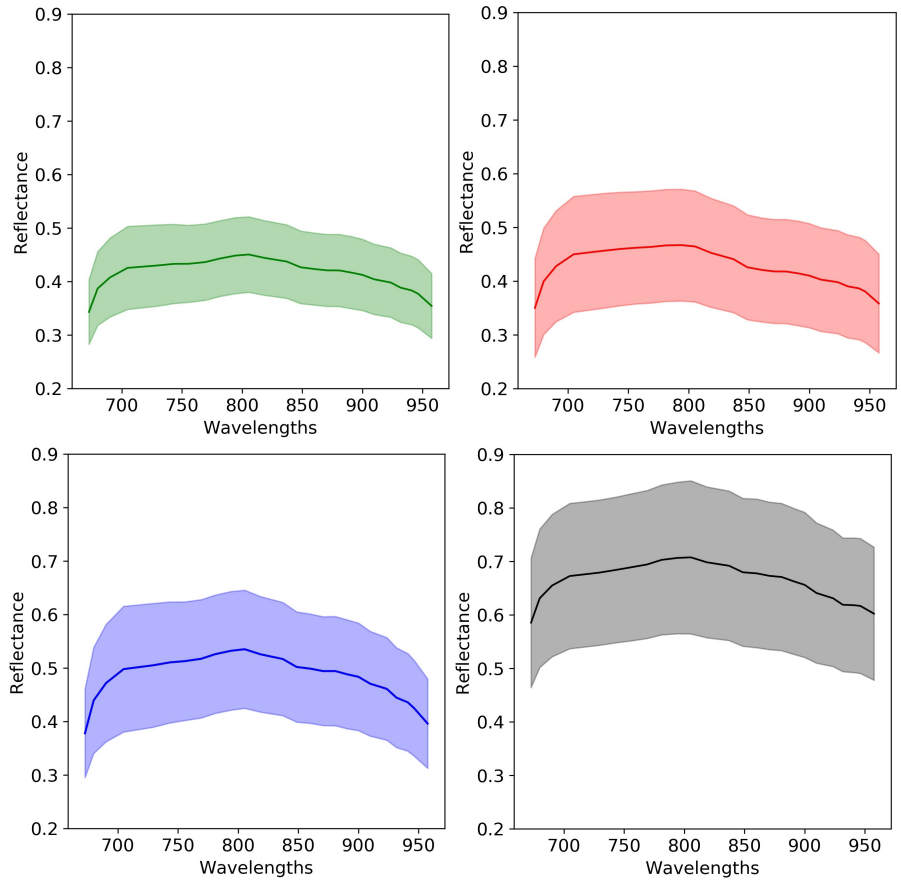


Figure 5.9: Variation of spectral information in snapshot NIR images of red-meat types. Colours Green, Red, Blue and Black show the mean spectrum and the variation around the mean for Lamb, Beef, Pork and Fat respectively

For further investigation, the variation spectral information within each class was analysed to show the impact of light scattering on spectra of each meat type. The standard deviation was used to quantify the variation in spectra within each class. The standard deviation was computed to indicate that the samples tend to be close to or far from the mean spectrum. As an example, we used NIR snapshot data to show the variation analysis. Figure 5.9 shows the variation of spectral information within each class in snapshot NIR images.

The results in Figure 5.9 show that the spectra of classes are highly overlapped, which shows the challenge in the classification task between these classes. For example, the spectra of lamb and beef are totally overlapped and located in the same range of reflectance values. Thus, the distance (like the Euclidean distance) measure between any two samples from these classes will be very small and does not represent the difference between these samples. The average standard deviation across all wavelengths for lamb, beef, pork and fat are 0.07, 0.10, 0.11 and 0.14, respectively. The lamb class has the lowest standard deviation, which means that the samples within this class are more similar and closer to their mean spectrum.

5.4.2 Line-scanning HSI

The training set of line-scanning HSI images was used to train all models by using the training data items as shown in Table 5.2. In the case of the PLS–DA and SVM models, a window of 5×5 was cropped around each item. Then, the mean spectra of these windows were computed for obtaining the raw spectral data of each class, where the size of each spectrum is (1×225). For training the SVM model, raw spectral data were used with the cross-validation for best model selection. In training the PLS–DA model, we implemented the same setting as in [79]. Thus, the second derivative of raw spectra was extracted by using the SG method [69] with a window of 9 and second order polynomial fitting; cross-validation was used for optimizing the best number of components of the PLS–DA model.

In the proposed 3D–CNN model for line-scanning HSI, parameters were chosen empirically based on numerous experiments for achieving high accuracy and a light-weight model. The spatial size (i.e., the size of the 3D window) is a critical parameter affecting the performance of the proposed CNN model. Figure 5.10 shows the influence of spatial window size on the validation set. The spatial size of 5×5 was selected as the optimum value of the spatial size of the model. Table 5.3 shows all details of the proposed architecture for line-scanning HSI. The optimised model architecture (i.e., with spatial size of 5×5) was converged well and showed a stable convergence in both training and validation, as shown in Figure 5.11.

The baseline models and the proposed 3D–CNN were applied to the test set of images. Table 5.4 shows the evaluation measures for each model. The results show that the 3D–CNN model achieved significant enhancements in comparison with other models in the case of mean F_1 score, average accuracy and overall accuracy. This significant enhancement by using the 3D–CNN model shows the importance of combining spectral and spatial features into a single model. The SVM model with RBF kernel shows competitive results in comparison with the PLS–DA model.

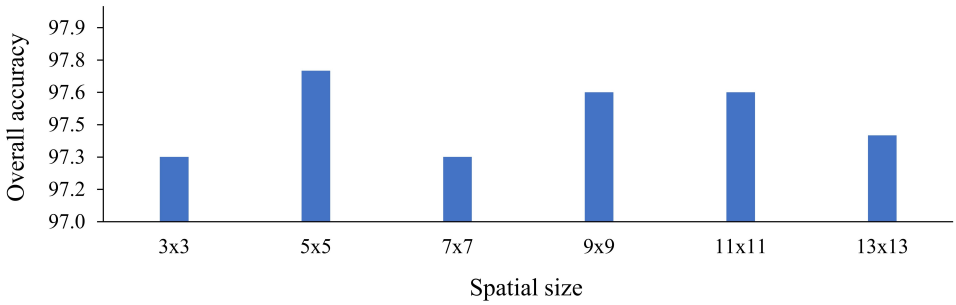


Figure 5.10: Impact of spatial size on the accuracy of the proposed 3D–CNN for line-scanning HSI classification.

Table 5.3: Architecture and specifications of the proposed 3D–CNN model for line-scanning HSI classification.

Layer	**Kernel**	**Output size**	**Stride**	**Activation**
Input	–	5×5× 225	–	–
3D-Conv 1	(3,3,15)	4@(5×5×225)	(1,1,1)	ReLU
3D-Conv 2	(3,3,15)	4@(5×5×225)	(1,1,1)	ReLU
3D-Max-pool 1	–	4@(3×3×75)	(2,2,3)	–
3D-Conv 3	(3,3,5)	8@(3× 3×75)	(1,1,1)	ReLU
3D-Conv 4	(3,3,5)	8@(3×3×75)	(1,1,1)	ReLU
3D-Max-pool 2	–	8@(2×2×25)	(2,2,3)	–
Flatten	–	1 × 800	–	–
Fully connected	–	1 × 128	–	ReLU
Output	–	1 × 4	–	softmax
The total number of trainable parameters is 110,088				

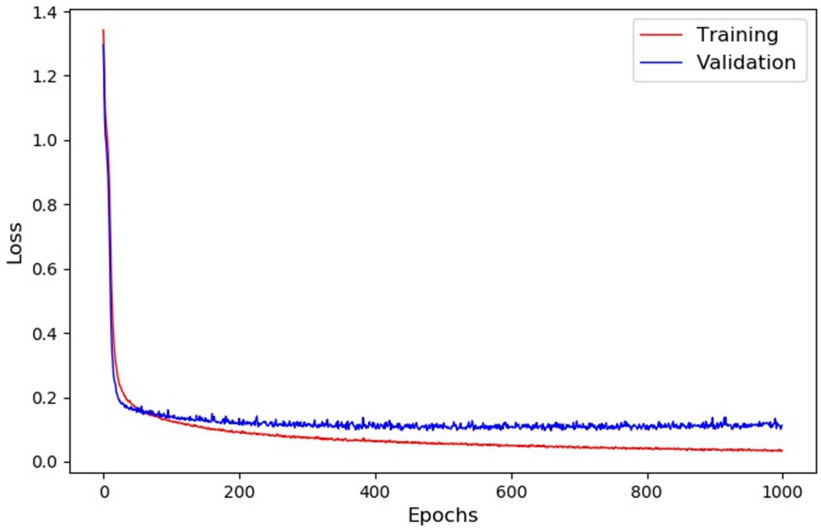


Figure 5.11: Learning curve of the proposed 3D–CNN model for line scanning HSI classification.

Table 5.4: Performance evaluation of the proposed 3D–CNN model, in comparison with PLS–DA and SVM for red-meat classification by using line-scanning HSI imaging system.

Model	**F_1 score**				**Mean F_1**	**A.A**	**O.A**
	LAMB	BEEF	PORK	FAT			
PLS–DA	90.1	95.2	94.8	89.7	92.5	92.4	93.2
SVM	94.7	98.5	95	94.2	95.6	95.4	96.3
3D–CNN	97.8	99.8	97.8	97.4	98.2	98.1	98.6

As a qualitative evaluation, Figure 5.12 provides classification maps of a selected set of images from the test images. The classification maps of the 3D–CNN model show that this model provides accurate predictions of classes, while the edges between materials are efficiently preserved. These accurate classification maps reflect the robustness of this model for the task of material classification, while the maps of base-line models show many miss-classified pixels in all types of red meat materials.

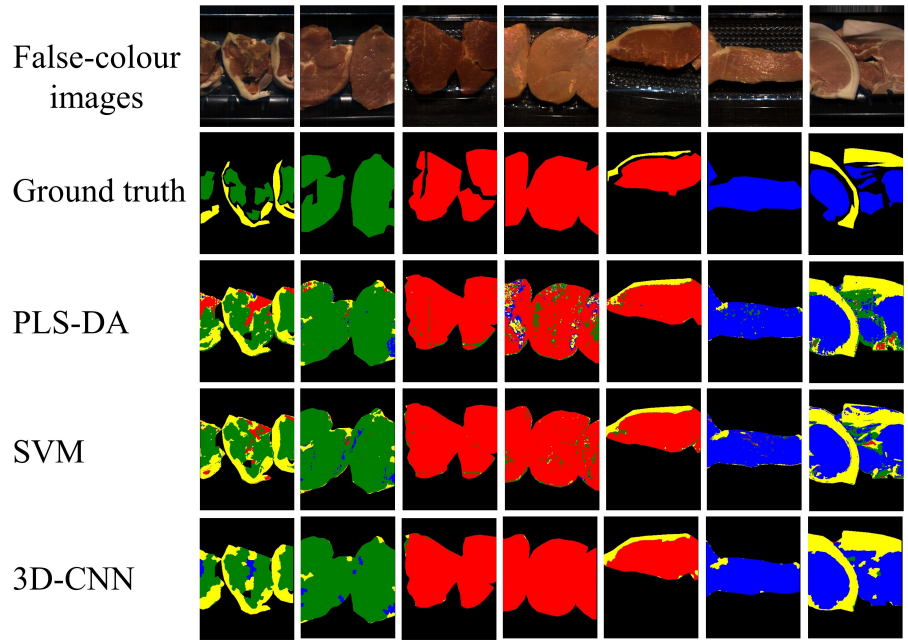


Figure 5.12: Classification maps of PLS–DA, SVM and 3D–CNN models of line-scanning images of meat samples; colours Green, Red, Blue and Yellow represent the classes LAMB, BEEF, PORK and FAT, respectively. Left to right: Columns 1–2, 3–5 and 6–7 show images of lamb, beef and pork, respectively.

For more investigation, Table 5.5 shows the performance of the proposed 3D–CNN by all data and average image evaluations. The results show that accuracy is approximately equal in both cases, which means that the model provided stability and applicability for any real-life applications. Also, Table 5.5 shows the efficiency of the proposed postprocessing method (weighted superpixel-based prediction) in comparison with standard pixel-based and superpixel-based prediction methods. The results show that considering superpixel segments does not add any enhancements to the traditional pixel-based prediction method. While the proposed weighted superpixel-based method significantly enhanced the classification accuracy of the 3D–CNN model.

Figure 5.13 provides a visual comparison between weighted superpixel-based, pixels-based and superpixel-based prediction methods. The weighted superpixel-based method showed very accurate classification, where many of the misclassified pixels were successfully corrected. Moreover, the edges and the shapes of the samples were preserved efficiently.

Table 5.5: Evaluation of the proposed 3D–CNN model for line-scanning HSI: In the case of all data and image-based evaluation, pixel-based prediction, superpixel-based prediction and the proposed weighted superpixel-based prediction method.

Prediction mode	Class	Avg images			All data		
		F_1	A.A	O.A	F_1	A.A	O.A
Pixel-based	LAMB	98.3	97.6	97.9	95.8	97.3	97.6
	BEEF	99.4			99.2		
	PORK	98.2			97		
	FAT	95.6			97.1		
Superpixel-based	LAMB	98	97.4	97.7	95.7	97.1	97.4
	BEEF	99.3			99		
	PORK	98.2			96.9		
	FAT	95.3			96.4		
Weighted superpixel-based	LAMB	98.8	98.1	98.7	97.8	98.1	98.6
	BEEF	99.8			99.8		
	PORK	98.5			97.8		
	FAT	96.5			97.4		

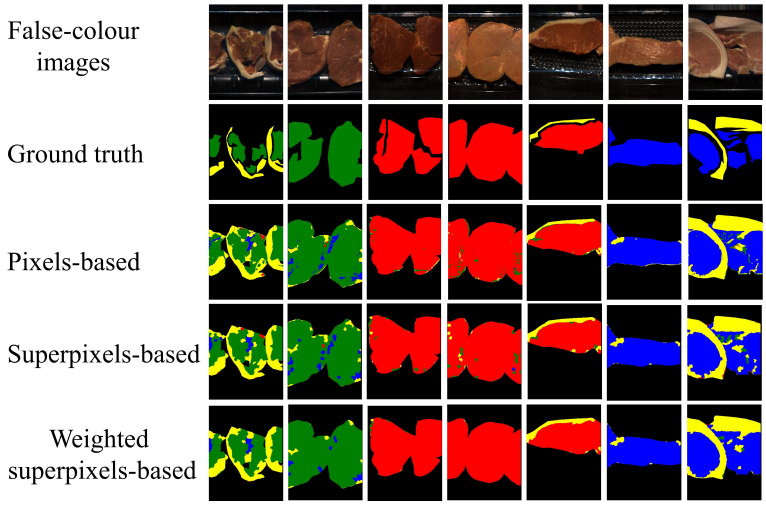


Figure 5.13: Classification maps of the 3D–CNN model for line-scanning images, using different prediction methods: Pixel-based, superpixel-based, and weighted superpixel-based; colours Green, Red, Blue and Yellow represent the classes LAMB, BEEF, PORK and FAT, respectively. *Left to right*: Columns 1–2, 3–5, and 6–7 show images of lamb, beef and pork, respectively.

5.4.3 NIR snapshot HSI

In NIR snapshot HSI, baseline models were implemented by using the training data items of NIR snapshot training images. For each item, a window of 5×5 was cropped around each item. The mean spectrum was computed for obtaining the raw spectral data, where each spectrum has a size of (1×25). The SVM model was trained on the raw spectral data. In the case of the PLS–DA model, a window size of 5 and second order polynomial fitting were used while applying the SG method [69] for computing the second derivative spectra of the raw data. Next, a PLS–DA model was trained on the second derivative spectral data. For both models, the cross-validation approach was used for obtaining the best parameters of each model.

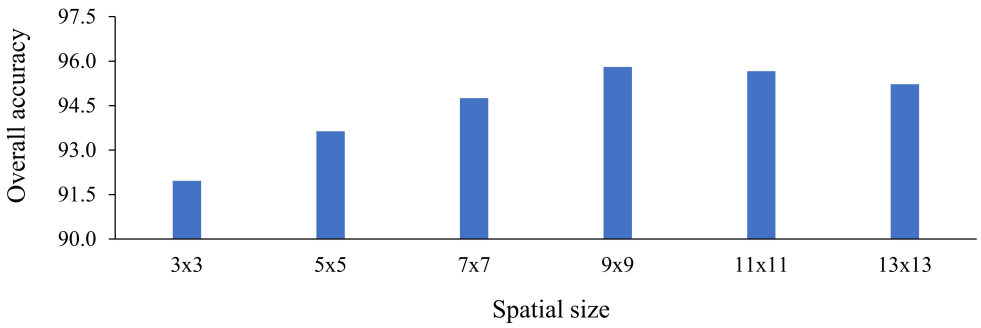


Figure 5.14: Impact of spatial size on the accuracy of the proposed 3D–CNN for NIR snapshot HSI classification.

In the proposed 3D–CNN model, the same items were used to extract a set of $W_s \times W_s \times 25$ windows around these items. Next, these windows (including a set for training and another set for validation) were used as input of the model. For fair comparison between the three HSI systems, the same architecture as for line scanning was used with the following differences: (1) The input size (i.e., the spatial size) was empirically optimized to $9 \times 9 \times 25$; Figure 5.14 shows the effect of changing the spatial size on the model accuracy on the validation set. (2) Kernel representations: $(5, 5, 5)$ and $(3, 3, 3)$ for first and second 3D-Conv blocks, respectively. (3) Strides of $(2, 2, 2)$ for each 3D-max-pooling layer. Thus, there are $70,272$ trainable parameters in total. Table 5.6 shows the specifications of the optimized architecture (i.e., with spatial size of 9 × 9) of the proposed 3D–CNN model for NIR snapshot HSI classification. In the training phase, the model was converged well and showed a stable convergence in both training and validation, as shown in Figure 5.15.

Table 5.6: Architecture and specifications of the proposed 3D–CNN model for NIR snapshot HSI classification.

Layer		Kernel	Output size	Stride	Activation	Dropout
1	Input	–	9×9×25	–	–	–
2	3D-Conv 1	(5,5,5)	4@(9×9×25)	(1,1,1)	ReLU	–
3	3D-Conv 2	(5,5,5)	4@(9×9×25)	(1,1,1)	ReLU	–
4	3D-Max-pooling 1	–	4@(5×5×13)	(2,2,2)	–	0.25
5	3D-Conv 3	(3,3,3)	8@(5×5×13)	(1,1,1)	ReLU	–
6	3D-Conv 4	(3,3,3)	8@(5×5×13)	(1,1,1)	ReLU	–
7	3D-Max-pooling 2	–	8@(3×3×7)	(2,2,2)	–	0.25
8	Flatten	–	1×504	–	–	–
9	Fully connected	–	1×128	–	ReLU	0.25
10	Output	–	1×4	–	Softmax	–
The total number of trainable parameters is 70,272						

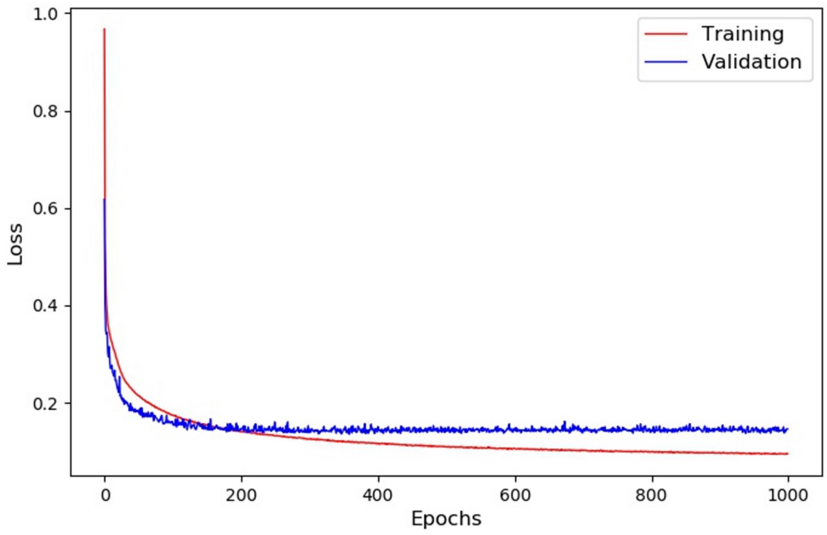


Figure 5.15: Learning curve of the proposed 3D–CNN model for NIR snapshot HSI classification.

The evaluation measures of all models for the testing images are shown in Table 5.7. The results show that the 3D–CNN model outperforms the baseline models regarding accuracy and F_1 scores for all classes. The proposed 3D–CNN model achieved high F_1 scores for all classes, which reflects the balance in recall and precision of each class. Figure 5.16 provides a visual comparison between the baseline models and the proposed 3D–CNN model for NIR snapshot classification. As shown in the figure, the 3D–CNN model accurately classified the types of meat, and the edges between fat and meat are maintained efficiently.

Table 5.7: Performance evaluation of the proposed 3D–CNN model in comparison with PLS–DA and SVM for red-meat classification by using NIR snapshot HSI imaging system.

Model	**F_1 score**				**Mean F_1**	**AA**	**O.A**
	LAMB	BEEF	PORK	FAT			
PLS–DA	80.5	88.1	87.1	76.5	83.1	81.2	84.2
SVM	86.1	93.9	89.6	91.8	90.4	90.9	90.1
3D–CNN	94.0	99.3	95.1	98.2	96.7	96.6	96.9

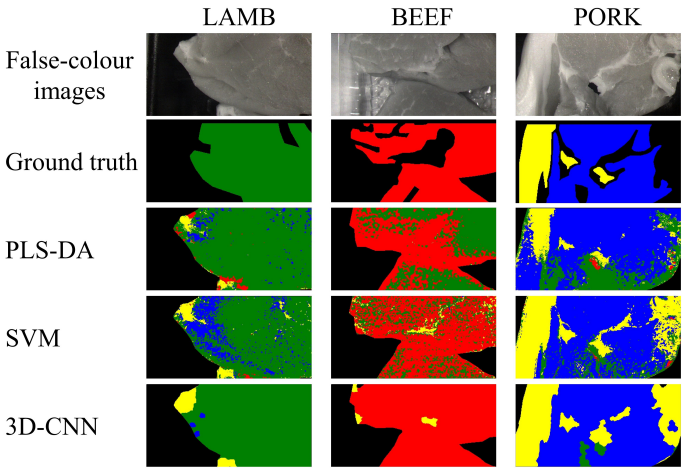


Figure 5.16: Classification maps of PLS–DA, SVM and 3D–CNN models of NIR snapshot images of meat samples. This visual comparison uses colours Green, Red, Blue and Yellow for classes LAMB, BEEF, PORK and FAT, respectively.

Table 5.8 provides the performance evaluation of the proposed 3D–CNN model with all data and average image evaluations. The results showed that the model is

stable and accurate when classifying each image individually. Also, Table 5.8 shows the robustness of the proposed postprocessing method on the NIR snapshot dataset, in comparison with standard pixel-based and superpixel-based classifications. Figure 5.17 visualises the efficiency of the proposed weighted superpixel-based prediction method on selected NIR snapshot HSI images.

Table 5.8: Evaluation of the proposed 3D–CNN model for NIR snapshot HSI: In the case of all data and image-based evaluation, pixel-based prediction, superpixel-based prediction and the proposed weighted superpixel-based prediction method.

Prediction mode	**Classes**	**Avg images**			**All data**		
		F_1	**A.A**	**O.A**	**F_1**	**A.A**	**O.A**
Pixel-based	LAMB	93.2	93.7	93.8	93.5	96.1	95.8
	BEEF	99.1			98.3		
	PORK	92.8			94.6		
	FAT	96.6			97.6		
Superpixel-based	LAMB	93.4	93.7	93.8	90.6	94.9	95.3
	BEEF	99.1			98.4		
	PORK	92.7			93.3		
	FAT	96.3			97.2		
Weighted superpixel-based	LAMB	95.6	94.5	95.3	94	96.6	96.9
	BEEF	99.6			99.3		
	PORK	92.4			95.1		
	FAT	94.9			98.2		

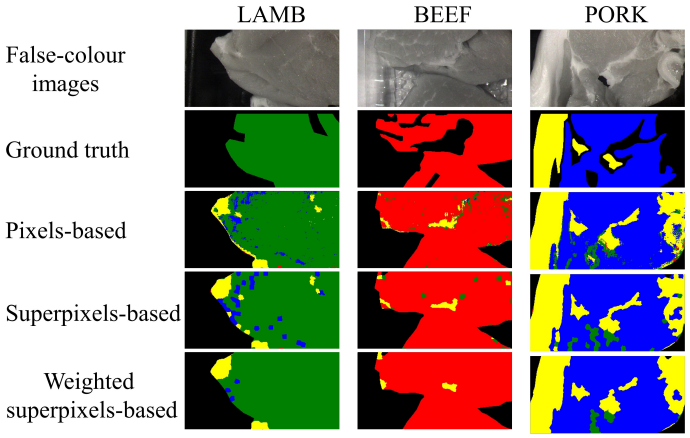


Figure 5.17: Classification maps of the 3D–CNN model for NIR snapshot images, using different prediction methods: Pixel-based, superpixel-based and weighted superpixel-based; colours Green, Red, Blue and Yellow represent classes LAMB, BEEF, PORK and FAT, respectively.

5.4.4 VIS snapshot HSI

In VIS snapshot HSI, baseline models were implemented by using the training items of VIS snapshot training images. For all items, the mean spectra of 5×5 windows, with a size of (1×16) for each spectrum, were computed for obtaining the raw spectra. Then, the SVM model was trained on the raw spectral data. In the case of PLS–DA models, a window size of 5 and second order polynomial fitting were used, by employing the SG method [69] for computing the second derivative spectra of the raw spectral data. Then, the PLS–DA model was trained on second derivative spectral data. For both models, the cross-validation approach was used for tuning the parameters of each model.

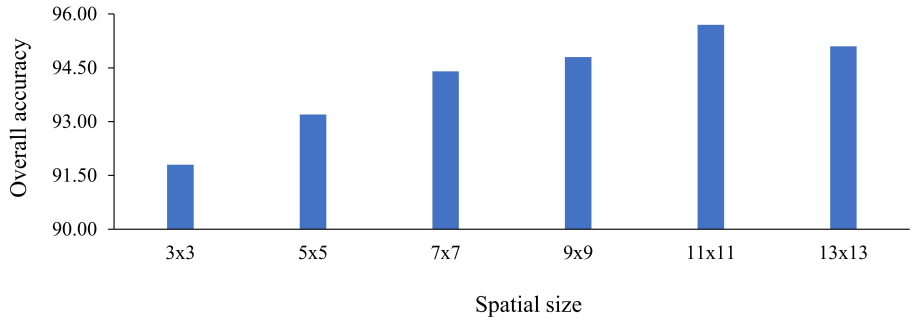


Figure 5.18: Impact of spatial size on the accuracy of the proposed 3D–CNN for VIS snapshot HSI classification.

In the proposed 3D–CNN model for VIS snapshot data, the same items were used to extract a set of $W_s \times W_s \times 16$ windows around these items. These windows (including a set for training and another set for validation) were used as input of the model. For fair comparison between the three HSI systems, the same architecture as in line scanning and NIR snapshot was used with the following differences: (1) The input size (i.e., the spatial size) was empirically selected to be $11 \times 11 \times \times 16$; Figure 5.18 shows the effect of changing the spatial size of the model's accuracy on the validation set. (2) Kernel representations: $(5, 5, 3)$ and $(3, 3, 3)$ for the first and second 3D-Conv blocks, respectively. (3) Strides $(2, 2, 2)$ and $(2, 2, 1)$ for the first and second 3D-max-pooling layers, respectively. Thus, $78,488$ is the total number of trainable parameters.

Table 5.9 shows the specifications of the optimised architecture (i.e., with spatial size of 11×11) of the proposed 3D–CNN model for NIR snapshot HSI classification. In the training phase, the model was converged well and showed a stable convergence in both training and validation, as shown in Figure 5.19.

Table 5.9: Architecture and specifications of the proposed 3D–CNN model for VIS snapshot HSI classification.

	Layer	Kernel	Output size	Stride	Activation	Dropout
1	Input	–	11×11×16	–	–	–
2	3D-Conv 1	(5,5,3)	4@(11×11×16)	(1,1,1)	ReLU	–
3	3D-Conv 2	(5,5,3)	4@(11×11×16)	(1,1,1)	ReLU	–
4	3D-Max-pooling 1	–	4@(6×6×8)	(2,2,2)	–	0.25
5	3D-Conv 3	(3,3,3)	8@(6×6×8)	(1,1,1)	ReLU	–
6	3D-Conv 4	(3,3,3)	8@(6×6×8)	(1,1,1)	ReLU	–
7	3D-Max-pooling 2	–	8@(3×3×8)	(2,2,1)	–	0.25
8	Flatten	–	1×576	–	–	–
9	Fully connected	–	1×128	–	ReLU	0.25
10	Output	–	1×4	–	Softmax	–
The total number of trainable parameters is 78,488						

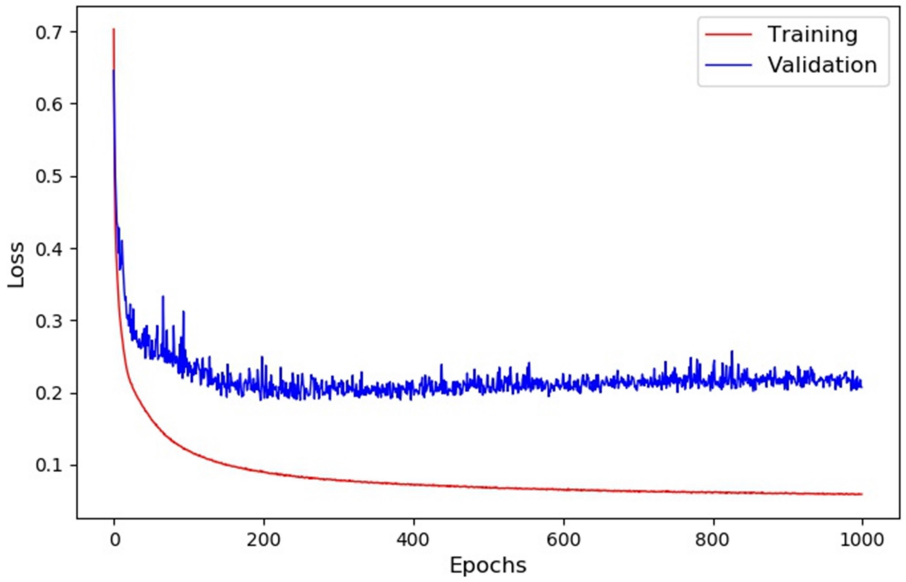


Figure 5.19: Learning curve of the proposed 3D–CNN model for VIS snapshot HSI classification.

After training the models, the test set of VIS snapshot images was used for obtaining the final evaluation. Table 5.10 shows a comparison between the results of PLS–DA, SVM and the proposed 3D–CNN model. The results show that the 3D–CNN model outperformed the others, despite the limitation in spectral information in VIS snapshot HSI. The 3D–CNN model provided a high accuracy and average F_1 score, which reflects the importance of spatial information in this kind of HSI systems. A visual comparison between the baseline models and the 3D–CNN model is provided in Figure 5.20. Classification maps, as in Figure 5.20, show that 3D–CNN successfully classified all the samples in a clear and efficient way, while the baseline model produced many misclassified pixels.

Table 5.10: Performance evaluation of the proposed 3D–CNN model, in comparison with PLS–DA and SVM for red-meat classification by VIS snapshot HSI imaging system.

Model	**F_1 score**				**Mean F_1**	**AA**	**O.A**
	LAMB	BEEF	PORK	FAT			
PLS–DA	67.3	81.7	75.6	83.1	76.9	77.2	77.9
SVM	87.7	93.5	91.9	91.3	91.1	91.6	91.6
3D–CNN	96.3	98.1	96.8	95.5	96.7	96.5	97.1

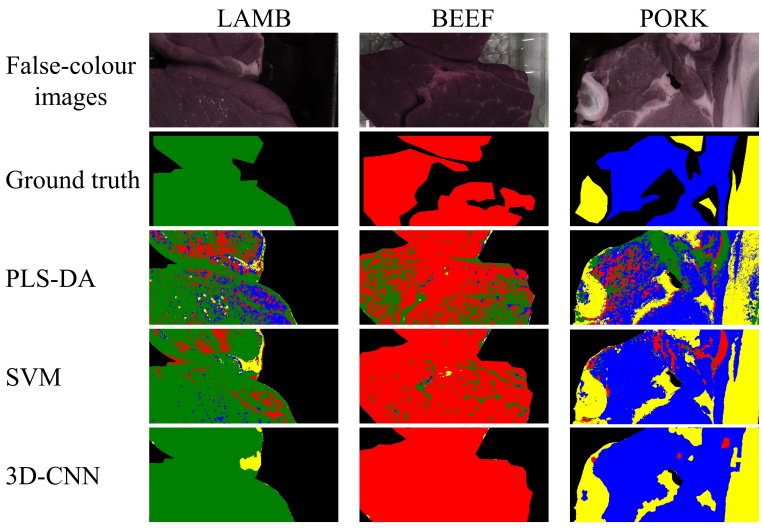


Figure 5.20: Classification maps of PLS–DA, SVM and 3D–CNN models of VIS snapshot images of meat samples. This visual comparison uses colours Green, Red, Blue and Yellow for classes LAMB, BEEF, PORK and FAT, respectively.

Table 5.11: Evaluation of the proposed 3D–CNN model for VIS snapshot HSI: In the case of all data and image-based evaluation, pixel-based prediction, superpixel-based prediction, and the proposed weighted superpixel-based prediction method.

Prediction mode	**Classes**	**Avg images**			**All data**		
		F_1	**A.A**	**O.A**	**F_1**	**A.A**	**O.A**
Pixel-based	LAMB	93.5	94.3	93.9	93.8	95.3	95.8
	BEEF	97.7			97.2		
	PORK	96.9			95.6		
	FAT	95.4			95		
Superpixel-based	LAMB	93.8	94.1	93.7	93.7	95.1	95.6
	BEEF	97.8			97.1		
	PORK	96.7			95.5		
	FAT	94.4			94.7		
Weighted superpixel-based	LAMB	95.4	95.4	94.9	96.3	96.5	97.1
	BEEF	97.7			98.1		
	PORK	97.3			96.8		
	FAT	95.2			95.5		

The 3D–CNN model for VIS snapshot HSI was also evaluated with all data and average image evaluations. The results, as given in Table 5.11, show that the 3D–CNN model performs well in the case of image-based and all data evaluations. Also, in VIS snapshot images, the proposed postprocessing method shows an efficiency on this HSI type. Table 5.11 shows that the proposed postprocessing method significantly enhanced the accuracy of the 3D–CNN model. For a visual comparison between the three prediction modes of the 3D–CNN model on VIS snapshot HSI images, Figure 5.21 provides the classification maps of the 3D–CNN model at each prediction method.

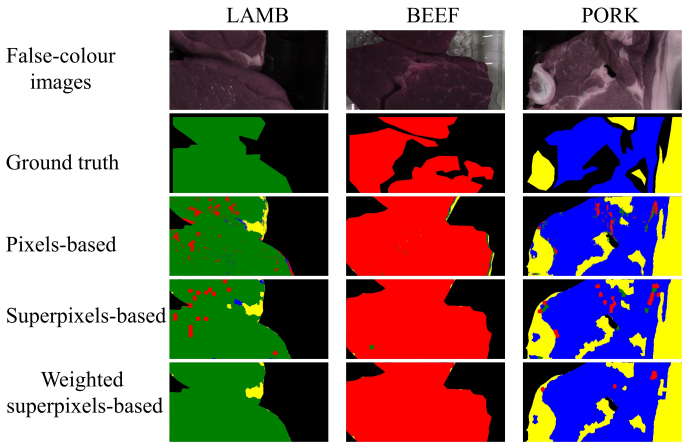


Figure 5.21: Classification maps of the 3D–CNN model for VIS snapshot images, using different prediction methods: pixel-based, superpixel-based and weighted superpixel-based; colours Green, Red, Blue and Yellow represent classes LAMB, BEEF, PORK and FAT, respectively.

5.5 Analysis and discussion

The experimental results of red-meat classification models, for each HSI system, showed that the proposed 3D–CNN model achieved the highest efficiency, in comparison with PLS–DA and SVM models, in terms of per-class F_1 score, average F_1 score of all classes, average accuracy and overall accuracy. In line-scanning HSI, the 3D–CNN model achieved 98.6% overall accuracy compared to 96.3% and 93.2% for SVM and PLS–DA, respectively. In NIR snapshot HSI, the 3D–CNN model achieved 96.9% overall accuracy compared to 90.1% and 84.2% for SVM and PLS–DA, respectively. In VIS snapshot HSI, the 3D–CNN model achieved 97.1% overall accuracy compared to 91.6% and 77.9% for SVM and PLS–DA, respectively.

Figure 5.22 summarizes the overall accuracies of all models on each HSI system. The general overview of the figure shows that both the PLS–DA and SVM models are highly affected by the amount of spectral information; the accuracy significantly decreased on snapshot HSI data. For example, PLS–DA achieved an overall accuracy of 93.2%, 84.2% and 77.9% for line-scanning, NIR snapshot and VIS snapshot, respectively, where these HSI systems have a number N_{bands} of wavelengths of 225, 25, 16 for line scanning, NIR snapshot, and VIS snapshot, respectively. This variation in results shows that PLS–DA and SVM models with spectral features provide competitive results on low spectral resolution systems like line-scanning HSI, while for snapshot HSI, these models are insufficient for handling these challenging HSI data.

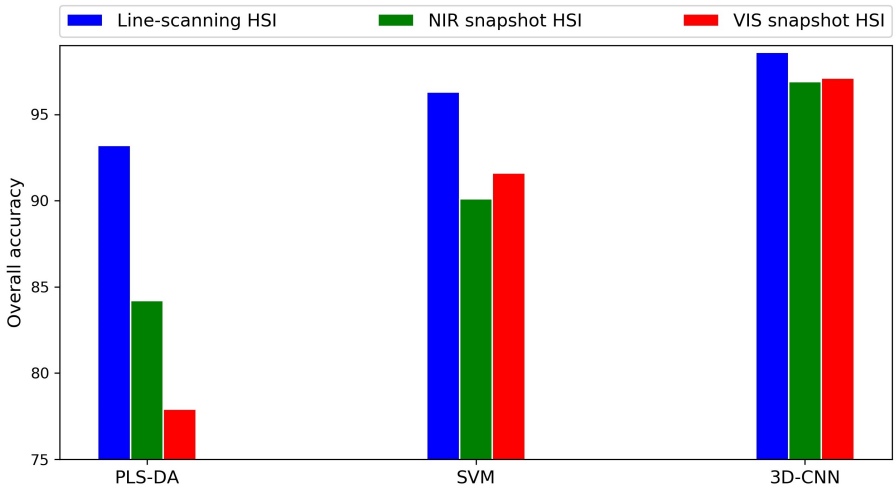


Figure 5.22: Summary of overall accuracies of PLS–DA, SVM and the proposed 3D–CNN model on each HSI system.

The proposed 3D–CNN model, as shown in Figure 5.22, appears to be an applicable model for all the considered HSI system types. The variation in the overall accuracy for all HSI systems significantly decreased, and the model performed well on all HSI systems with an overall accuracy of 98.6%, 96.9%, 97.1% for line scanning, NIR snapshot and VIS snapshot, respectively. These results show the robustness of the 3D–CNN model for using both spectral and spatial features in an efficient way. These observations show that joining both spectral and spatial features in a single model like 3D–CNN has a significant impact on the accuracy of an HSI classification model for material classification tasks.

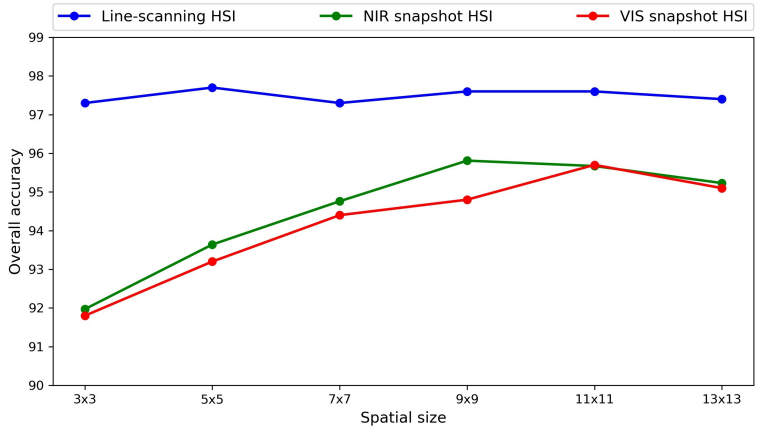


Figure 5.23: Influence of changing the spatial size on the overall accuracies of the proposed 3D–CNN model on each HSI system.

The spatial size (i.e., the size of the 3D window) is one of the main critical parameters in the proposed 3D–CNN model; it defines the amount of spectral and spatial features that the model takes into account. Figure 5.23 summarized the influence of this parameter on each HSI system. In line-scanning HSI, the overall accuracies over different spatial sizes appears to form a "smooth curve", which means that increasing the size of the 3D windows does not provide significant enhancements on accuracy. This stability regarding accuracy corresponds to the number of spectral features in line-scanning HSI systems.

In snapshot HSI systems, the impact of the spatial size shows a trade-off in results with optimal values for each NIR and VIS snapshot HSI. The 3D–CNN model for NIR snapshot data provides minimum overall accuracy for window size $(3, 3, 25)$ and optimal accuracy for window size $(9, 9, 25)$. In the case of VIS snapshot data, the model provides the same behavior, as for NIR data, with an optimal window size of

$(11, 11, 16)$. Thus, we note that the achieved accuracy at the size of $(11, 11, 16)$ is due to reduced spectral features of VIS snapshot HSI data (16 wavelengths). Thus, we observe that if the spectral information decreases, the need for more spatial features increases for accurate classification.

The proposed postprocessing method (i.e., weighted superpixel-based) for enhancing the prediction of the 3D–CNN model added significant enhancements for overall accuracy, per-class F_1 scores and average accuracy (as shown in Tables 5.5, 5.8 and 5.11) by using average images and all data evaluations. Using the proposed postprocessing method, in comparison with the standard prediction method (i.e., pixel-based), the 3D–CNN model achieved an improvement on the overall accuracy of 1%, 1.1% and 1.3% for line scanning, NIR snapshot and VIS snapshot, respectively. These enhancements show the flexibility of the 3D–CNN model for a deep use of spatial information and an ability for correct classification of specular pixels resulting from light-source effects. Visually, Figures 5.13, 5.17 and 5.21 show the efficiency of the proposed postprocessing method for the 3D–CNN model of line-scanning, NIR snapshot and VIS snapshot HSI images, respectively.

We also investigated features that were learned by the 3D–CNN model. Randomly, we selected a set of patches (i.e., a set of 3D windows around the extracted data items) from the validation set of NIR snapshot HSI. Then, we projected these patches into the models and extracted the computed output of the fully-connected layer. Extracted feature vectors (of size 128) were then fitted on the PCA models to reduce the dimensions. For the same patches, we extracted the mean spectra of the patch sets and applied other PCA models.

For a visual illustration, we plotted the first two components of all the PCA models. Figure 5.24, left, presents the PCA scatter plots of the raw HSI data, while on the right is the extracted 3D–CNN features. Clearly, Figure 5.24, right, shows that the 3D–CNN model is able to convert raw spectral-spatial data into a useful representation with a very good separation between the classes in the PCA space, while in the case of the original spectral data, see Figure 5.24, left, class regions are highly overlapping and look only like one or two clusters.

Also, we note that the explained variance ratios of the PCA models, as shown in Fig 5.25, right, for NIR snapshot data as an example of the extracted 3D–CNN features, are significant for many components (i.e., not only the first and second components). For example, the first 2 PCA components of the 3D–CNN features present only 29.9% out of 100% of the total variance of the data, which means that there is still significant information in the renaming components. While in the PCA model of raw data, the first component has the most significant explained variances (equal to 99.3% out of 100% in the case of NIR snapshot data), as shown in Fig 5.25, left.

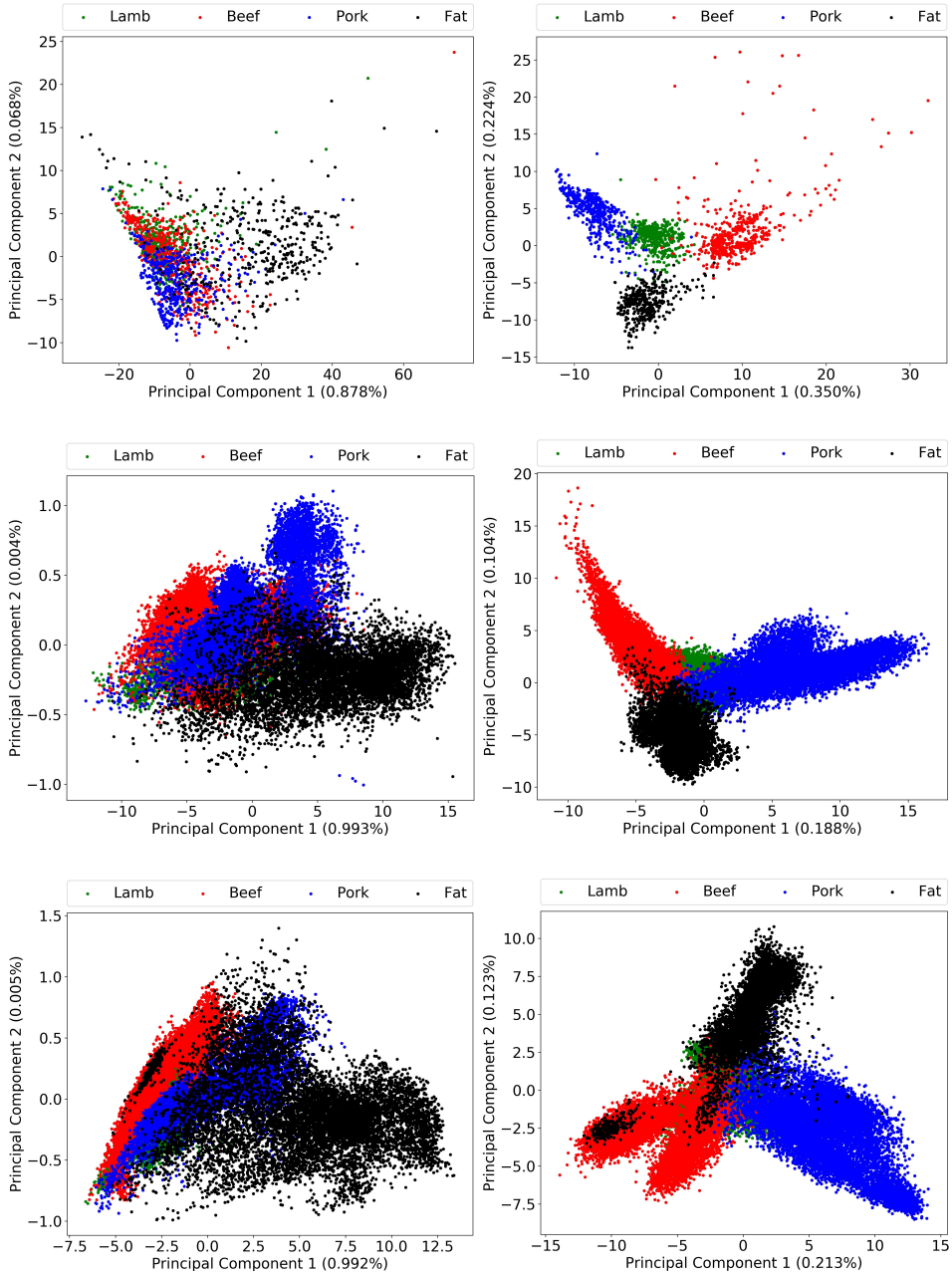


Figure 5.24: PCA analysis: PCA scatter plots for visualizing the separation between classes. *Left*: First two PCA components of the original spectral features. *Right*: First two PCA components of the learned features extracted by 3D–CNN models. Rows 1–3 represent PCA analysis for line-scanning , NIR snapshot and VIS snapshot HSI data, respectively.

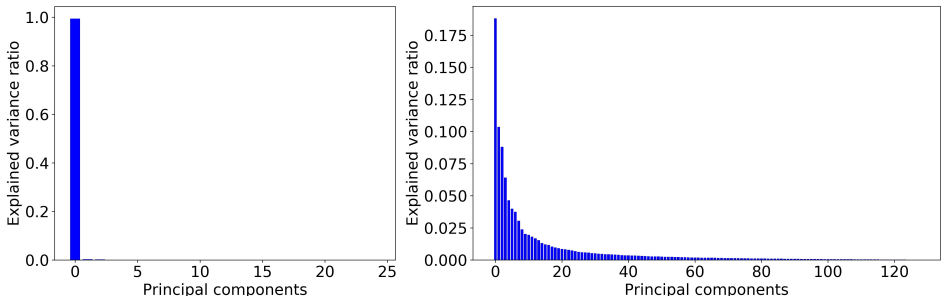


Figure 5.25: Explained variances of PCA models of NIR snapshot data. *Left*: Explained variance of the PCA model of raw spectra. *Right*: Explained variance of the PCA model of the learned features extracted by 3D–CNN models.

This observation means that the 3D–CNN features of meat are well distributed and clustered in the feature space, and they can be visualised with other PCA components rather than just first and second components. This analysis shows that the proposed 3D–CNN model is an efficient tool for the classification of challenging snapshot HSI data, and for understanding the patterns of each type of meat.

Due to the portability of HSI snapshot cameras (i.e., a completely portable device or mobile HSI system), we evaluated the time for image-classification (i.e., for classifying a single NIR snapshot image of 216×409) for the investigated models. Results show another efficiency of the 3D–CNN model, where the 3D–CNN model was 4.7 times faster than SVM and 2.9 times faster than the PLS–DA model; the classification times are 13, 38, 61 seconds for 3D–CNN, PLS–DA and SVM, respectively, running on the same machine.

The execution time analysis shows that the 3D–CNN model performs better than the baseline model. However, the achieved results are still not valid for real-time requirements (as discussed in Chapter 1). Thus, the 3D–CNN is efficient in terms of accuracy and understanding the spatial and spectral information of HSI images, while the execution time of the model raises an insufficiency regarding any real-time implementation for the meat industry. In fact, the time-consuming task in the 3D–CNN model is the used pixel-wise classification processes. These processes are independent and repeated for each pixel in an image to be classified. Thus, implementing these processes in parallel form could efficiently improve the execution time of the model, and solving the issue in this way could be considered as a further research direction. Moreover, adapting the proposed 3D–CNN architecture to classify all pixels in one shot (i.e., the input of the model is the whole image instead of the target pixel and the region around it) also is suggestive of a research opportunity to further investigate reducing the execution time of an HSI image.

Snapshot HSI cameras are designed to support the collection of spectral data at the video rate. The proposed 3D–CNN shows efficiency in accurately classifying snapshot images of a sequence HSI images (HSI video). Figure 5.26 shows a sequence of snapshot HSI of beef samples; images were collected at a rate of 8 frames per second (i.e., 8 hypercubes/sec).

As shown in Figure 5.26, for both NIR and VIS snapshot sequences, the 3D–CNN model accurately classified all sample portions. Thus, with regard to accuracy, the model is efficient and supports snapshot HIS video classification, while more attention needs to be paid to time computation in further research that investigates these kinds of HSI imaging systems.

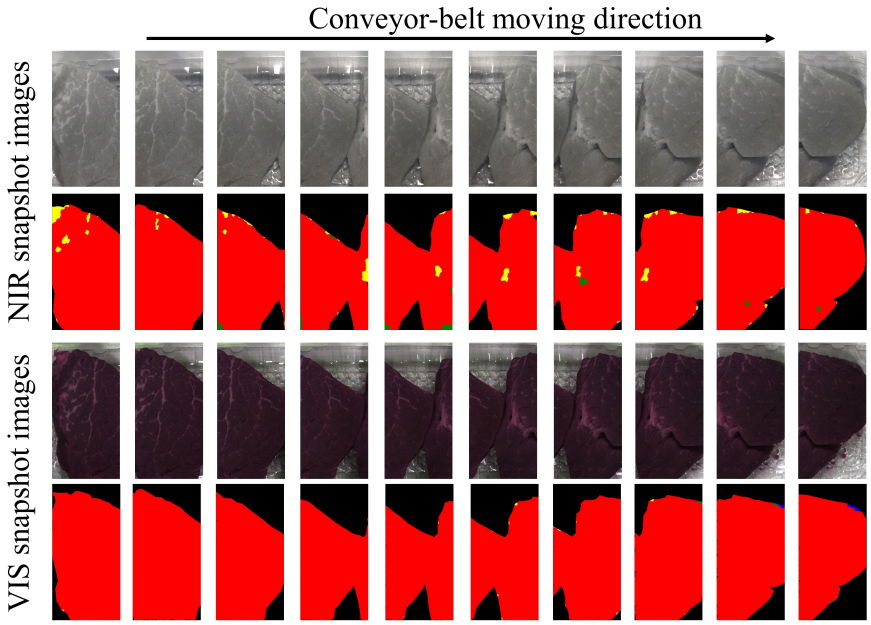


Figure 5.26: Classification maps of NIR and VIS snapshot HSI sequences resulting from the proposed 3D–CNN model. Colours Red and Yellow represent classes BEEF and FAT, respectively.

5.6 Summary

Material classification is an essential task in computer vision with a wide diversity of applications such as food processing, environmental monitoring and material sorting. HSI systems provide additional information for each material in the image, in

comparison to standard RGB imaging. Snapshot HSI systems are introduced as an intermediate solution between line-scanning HSI and RGB imaging, with the advantages of portability and working at the video rate.

Red-meat identification and authentication are important tasks in the meat industry. In this study, we investigated the potential and robustness of snapshot HSI systems, which provide limited spectral information for red-meat identification and authentication. Here, the red-meat classification problem is defined as a case study of a fine-grain material classification problem. We investigated the robustness of deep learning models (a 3D–CNN architecture) for classifying the type of meat. The quantitative and visual analysis clearly shows that the proposed deep 3D–CNN model outperforms PLS–DA and SVM models.

A comprehensive comparison between three HSI systems shows that the 3D–CNN model performs well on all HSI systems, while PLS–DA and SVM models are highly affected by the number of spectral features of an HSI system. Moreover, applying the 3D–CNN model to snapshot HSI systems (i.e., NIR and VIS) showed competitive results, compared with the standard line-scanning HSI, although their spectral information was limited. The proposed postprocessing method showed an efficiency on the performance of the 3D–CNN models of the three HSI systems. Results also show that the run-times of the implemented 3D– CNN model are much faster than for the other two models.

Visual results show that the 3D–CNN model converts snapshot HSI data into a useful presentation with an accurate separation between classes. The 3D–CNN model supports the implementation of a completely portable system (mobile HSI system) and a classification of video HSI. Thus, this research opens doors for further research towards real-time HSI classification and mobile HSI systems using snapshot HSI.

5.7 Links

This chapter focused on proposing a novel deep learning framework (including an architecture for classification and postprocessing method) for material classification by using joined spectral-spatial features of 3D–CNN networks. The proposed framework was evaluated against state-of-the-art models and on three different HSI systems. Moreover, fully joined spectral-spatial features, of fully-connected 3D–CNN networks, were evaluated in comparison with spectral features (i.e., raw spectra or derivative spectra). The proposed models and methods were tested on a fine-grain material classification problem (using red-meat discrimination as a case study).

Our results in this chapter showed interesting compatibility with the achieved results in Chapter 4, where considering the spectral-spatial features in the models significantly enhanced the accuracy of the models. Moreover, we observed that the need for these kinds of features (i.e., spectral-spatial features) is important when the spectral features are decreased such as in snapshot HSI systems. The 3D–CNN model, with joined spectral-spatial features of NIR and VIS snapshot HSI systems (which have limited spectra, e.g., 25 and 16 wavelengths for NIR and VIS, respectively), significantly enhances the accuracy of the models with spectral features only (i.e., PLS–DA and SVM with average spectra). While for line-scanning HSI systems (which have 225 wavelengths in the VIS and NIR regions), the results of all models (i.e., 3D–CNN model, PLS–DA and SVM) were very close with a minor improvement achieved by the 3D–CNN model. Thus, this chapter demonstrated, in addition to the results of Chapter 4, the importance of taking the joined spectral-spatial features into account.

One of the main objectives of this chapter is to investigate the robustness of snapshot HSI systems (i.e., NIR and VIS HSI). The results showed that these HSI systems are able to provide an excellent classification for materials even though they are limited in spectra, but they need a complex and robust model, such as the proposed 3D–CNN model. Thus, CNN networks (by 3D–CNN) can fully utilize this limited spectral information by joining both spectral and spatial features.

Based on the results in this chapter, we found that snapshot HSI systems are superior tools for material classification, given their portability and capacity to work at the video rate. Moreover, the CNN networks showed a robustness for intelligently utilizing the features (i.e., spectral and spatial) of snapshot HSI systems. Thus, in the next chapter, we will investigate the robustness of CNN networks and snapshot HSI systems (especially the NIR systems) for solving the object detection problem based on both spectral and spatial features. Foreign object detection (or foreign body detection) will be the case study for the object detection and localization problem by using NIR snapshot HSI systems.

Chapter 6

Foreign Object Detection in Meat Products

This chapter reports on the potentials of hyperspectral imaging (HSI) for object detection, especially on an application of foreign object detection (FOD) in meat products. A sequential deep learning framework is proposed by using region-proposal networks (RPN) and CNN networks. Two independent datasets of images, contaminated with many types of foreign materials, were used for training and testing the proposed model. Results show that the proposed RPN model outperforms a selected search method in terms of accuracy, efficiency and run-time. An FOD model based on RPN and 3D–CNN or a selected search with a 3D–CNN solved FOD with an average precision of 81.0% *or* 50.6%*, respectively. This study demonstrates opportunities when using hyperspectral imaging systems for real-time object detection by combining spectral and spatial features.*

This chapter is organized as follows. Section 6.1 provides an introduction, which includes the motivations and objectives of the research presented in this chapter. Section 6.2 reviews the state-of-the-art techniques which are related to this research. Section 6.3 describes the used datasets and the HSI imaging system. The proposed methods and models are demonstrated in Section 6.4.1. Section 6.6 provides the experimental results and analysis of the proposed framework. Section 6.7 summarizes the chapter.

6.1 Introduction

Food inspection and monitoring are essential processes in the modern food industry. The main task in these processes is to ensure that the products are safe, wholesome and comply with international standards and legislation [158, 159]. In the meat industry, there are two fundamental types of inspections, product-based and carcass-based inspections. Meat-product inspection processes include ensuring correct labelling and packaging and a prevention of recall incidents due to physical or microbial contamination. Foreign matter (i.e., foreign objects or foreign bodies), defined as a kind of physical contamination in meat products, can accidentally fall into/on meat products during processing and packaging. These objects could be glass, metal, paper or plastic objects [39, 158, 159].

Existing technologies for automated FOD in the food industry [158,159] include metal detectors, magnetic separators, optical sorting systems, microwave imaging, *nuclear magnetic resonance imaging* (NMRI), ultrasound systems, and X-ray imaging. Each technology is employed for a specific type of food and specific types of foreign objects, and each has advantages and drawbacks [158,159]. For example, X-ray imaging systems perform accurately for detecting metal and plastic objects. However, they have limitations regarding work-environment safety, large-scale foreign materials, high costs, and radiation emissions [158–162].

HSI systems are robust, rapid and non-destructive tools for presenting both the chemical composition and spatial distributions of materials; these distributions are presented as a 3D image volume [45]. Thus, HSI images are commonly used in many applications in the food industry, for example for predicting safety or quality attributes [45, 85, 163], biological contaminant detection [164] and detection of physical contamination [85].

In all of the above technologies for automated FOD tasks, there are several limitations and challenges such as dealing with shape, size, colour and types of foreign materials that may contaminate the products. Thus, the visual properties of these objects are not known in advance and can be composed of any kind of contamination/materials. Moreover, localising the detected objects is very important for automating the inspection processes [158]. Technologies such as X-rays, NMRI and ultrasound imaging can penetrate inside the sample, while conventional RGB and HSI imaging can only process the sample's surface. Moreover, RGB and HSI systems are considered as low-cost systems in comparison with X-rays, NMRI and ultrasound imaging systems [45,158].

As meat products are the exemplar of the research presented in this thesis, we use FOD problems in meat products for testing and evaluating the proposed general FOD framework for any food product. Meat processing, regarding the quality and safety of products, is gaining more attention in the meat industry and related research. The industry aims to prevent recall occurrences throughout the whole meat processing stages (starting from plant processing, distribution, sale and consumption) and to keep products free of physical contaminants that cause recalls to occur during the process [39].

In general, meat is considered to be one of the food types that has a high probability of recall occurrences [39]. Costs for preventing these incidents are high in the meat industry; inspections to avoid foreign matters in products are typically performed manually, and thus are subject to human error, increase labour costs and are time-consuming. As an emerging technology, snapshot HSI systems [49, 50] can be used for automating the detection process of foreign objects in meat products, which positively reflects on the accuracy, costs and speed of the whole inspection process.

This chapter aims to investigate the robustness of snapshot HSI systems for real-time detection and localisation of foreign objects (or bodies) that may contaminate food products. Moreover, we investigate the efficiency of supervised learning approaches for FOD in food products. The main objectives of this chapter are as follows:

- Propose and evaluate a novel sequential deep learning framework for FOD in meat products.
- Develop and evaluate a novel region proposal model for solving object detection tasks in HSI imaging.
- Develop and evaluate a 3D–CNN model for extracting features and classification of normal and abnormal regions in HSI images.
- Provide a comparison with the state-of-the-art methods for FOD.

6.2 Related work

FOD systems based on imaging follow one of two main approaches for implementing FOD detection models. In a supervised learning approach, prior knowledge about the expected foreign objects and their physical properties is needed for training the models [165–167]. An unsupervised learning approach (anomaly detection) requires an understanding of normal materials while training the models on the desired materials [168, 169].

Supervised learning approaches demonstrate robustness and efficiency for many FOD tasks such as FOD on airfield pavements [165]. The accuracy of such approaches depends on the used features for FOD and the representations of foreign materials which are taken into consideration [165,167]. Handcrafted image features, such as *scale invariant feature transform* [170], *histograms of oriented gradients* [171] and *local binary patterns* [172], are considered to be insufficient for dealing with complex images (i.e., complex normal and abnormal regions) for FOD tasks [165].

Recently, CNN-based models have gained much attention and show an efficiency for dealing with several tasks in computer vision research, such as image recognition [173] and object detection [120, 174, 175, 177, 179, 180]. Object detection applies, for example, in RPN networks or target-regression approaches. The RPN was proposed in faster R-CNN [175] and mask R-CNN [176] algorithms. In these algorithms, both region proposal and classification of sub-models were designed in a single model by sharing features of the RPN model [175, 176]. RPN-based algorithms show efficiency, both in terms of run-time and accuracy, in comparison with

algorithms following "traditional" (i.e., non-neural network) methods for generating region proposals [120,174], such as selective search [178] or sliding windows.

Target regression was used for localisation tasks in the *single shot Multi-Box detector* (SSD) [180] and in *You-only-look-once* (YOLO) [179] algorithms. Both SSD and YOLO algorithms showed better performance regarding run-time, but lower accuracy compared to RPN-based algorithms. Moreover, SSD and YOLO show limitations in the detection of small objects due to their design [165].

In [165], a multi-stage deep learning algorithm was proposed for FOD and included three steps: an RPN model for generating a set of candidates of objects; a spatial transformer network for rectifying the resulting candidates; and a CNN network for classifying candidates into object categories or background.

A comparison in [165] shows that sequential approaches (i.e., multi-stage algorithms) for FOD are better than FOD detection in a single algorithm such as faster R-CNN or SSD, where the comparison reports that the multi-stage algorithm [165] is faster and more accurate than faster R-CNN and SSD algorithms for this kind of object detection problem (i.e., FOD on airfield pavements).

6.3 Dataset and samples preparation

A collection of fresh red-meat samples was procured from two local supermarkets. The total number of samples is 184, including lamb (67), beef (73) and pork (44). All of the samples were chosen from loin and leg chops. The samples were chosen from eight different commercial products which are commonly found in New Zealand shops. The meat samples were randomly separated into 102, 40, 40 meat samples for training, validation and testing purposes.

As mentioned in Chapter 2, three HSI imaging systems were implemented for collecting the datasets to be used in this thesis: line scanning, NIR snapshot and VIS snapshot. In this chapter, the NIR snapshot system was selected to collect the dataset of FOD in meat products. In fact, NIR snapshot was selected for the following reasons: (1) In comparison with the line-scanning system, the NIR snapshot system is more applicable for any particular application for food industry, where it works at the video rate and the image can be collected at a stationary position. In line scanning, the sample needs to be moved for scanning, and the high dimensionality of the resulting hypercubes could affect the processing time of the proposed models and methods. (2) In comparison with the VIS snapshot system, the NIR snapshot system provides more spectral information (range 600 - 900 nm) regarding the chemical composition of materials, which means that the collected spectra are

independent of the colour properties of the materials. In the VIS snapshot system, however, the collected spectral information (range of 400 - 600 nm) is more about the visual properties of the materials. So, NIR snapshot is useful for our FOD application to obtain an FOD model invariant to the visual properties of foreign objects in meat products.

The used snapshot HSI systems for acquiring all HSI images which are used in this chapter are described and demonstrated in more detail in Chapter 2; all system setup, parameters and reflectance calibration are explained in detail for the used snapshot HSI system (Section 2.4.2). The NIR snapshot system consists of NIR snapshot camera [49,50], a set of illumination units, a controlled movable conveyor belt and a computer running image acquisition software.

Figure 6.1: *Top*: Examples of used foreign materials in training and validation images. *Bottom*: Set of materials used in testing images.

For simulating the FOD problem in meat products, several sets of materials were collected, then they were added to the surfaces of the meat samples. Thus, the meat samples were contaminated with these materials. These materials are categorised as glass, soft and hard plastic, transparent plastic, papers and metals. In all material categories, a wide range of colours was taken into account. Moreover, these materials were grouped into two sets: one set was considered in training and validation processes, and the other set was left for testing and evaluating the proposed methods only. 6.1 shows examples of the materials that were used for the validation and testing phases.

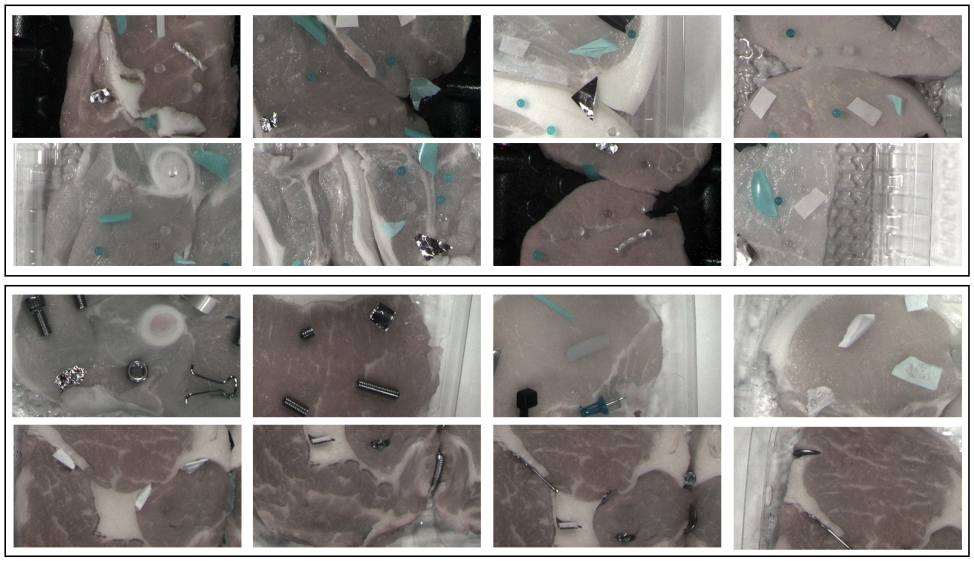


Figure 6.2: *Top*: Selected HSI images from the training and validation set. *Bottom*: Selected HSI images from the testing set. The images are false-colour images extracted from HSI images for visualization purposes.

The HSI system was used for collecting a set of $2,988$ snapshot HSI images, including 1,000 images of normal training meat samples (i.e., without adding any foreign materials), 937 images of contaminated training meat samples (i.e., contaminated by adding a random set of foreign materials), 250 images of normal validation samples, 220 images of contaminated validation samples and 581 images of contaminated testing meat samples. Figure 6.2 shows a selected set of images (from validation and testing sets) of meat products contaminated with foreign objects. After collecting the images, all images were calibrated and postprocessed, including illumination correction and normalisation steps, by using the method presented in Section 2.4.2.

6.4 The proposed framework for FOD detection

In this research, we aim to investigate the impact of spectral-spatial features for solving the object detection problem in HSI imaging and especially for FOD detection in meat products, where the proposed methods and models can be applied to other applications. A snapshot HSI camera is a robust tool for collecting spectral data at

the video rate; it is portable similar to RGB cameras with the advantage of having more spectral information of materials in scenes. This fact inspired us to investigate this tool for solving object detection for FOD in meat products. We believe this study is one of the first modern approaches to investigate the problem of object detection and localization by HSI imaging systems.

The proposed framework, as demonstrated in Figure 6.3, was inspired by methods and models presented in [120, 165, 173, 175]. The proposed framework is a combination of several modules in a sequential order. The framework consists of three modules: an RPN for generating a set of candidates with a probability of being foreign materials or normal materials; a *filtering module* for reducing the number of these candidates by having certain rules for regions with a high probability of being foreign materials; and a *classification module* for a final classification of the resulting top regions.

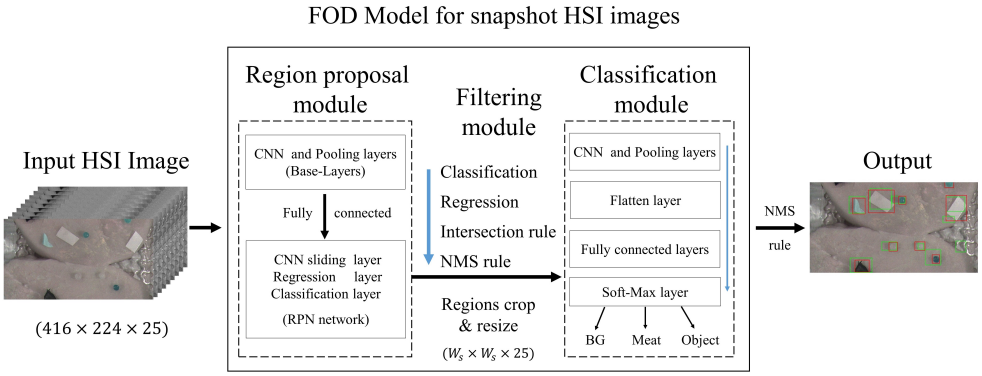


Figure 6.3: The proposed FOD model for meat products by using snapshot HSI images. The arrows present the sequential flow of the model.

A typical object detection framework in computer vision has the task of generating proposals as candidate regions for the object of interest that needs to be detected in the image. The number of these proposals is usually high, which reflects badly on the performance of the classification task as well as the performance of the final output of object detection framework. Thus, filtering these proposals based on certain criteria could help to improve the performance of the object detection framework. This filtering algorithm is called *non-maximum suppression* (NMS).

The main idea behind the NMS algorithm is to filter out the proposals based on their objectiveness scores (the probability of a region being an object in the image) and their overlapping by using the *intersection over union* (IoU) measure. Let B be a set of proposals (bounding box coordinates), S is a set of scores of each proposal and

D is the final set of proposals after the filtering process. The NMS algorithm starts with selecting M proposal that has the maximum score in S. Then, the algorithm removes M from the set B and puts it in the set D. For the M proposal, the algorithm removes any proposal from sets B and S which has an IoU value (overlapping) greater than a threshold τ. This process is repeated for all remaining boxes B. The final output of the NMS algorithm is the proposals in set D and their corresponding scores.

In the proposed FOD framework, the output of all modules was further post-processed by using the NMS algorithm as a final step. In the NMS step, a threshold of $t = 0.2$ was used to remove the overlapped bounding boxes around the detection. Thus, the remaining bounding boxes after the NMS step are the final output of the proposed FOD model for meat products. Figure 6.3 illustrates the proposed framework and its sequential flow.

6.4.1 RPN module

Deep learning requires a large set of labelled images. Transfer learning is commonly used for handling the issue of a small dataset by initialising models with an already-existing and well-trained model. Unfortunately, this approach is not applicable on HSI images due to their dimensions (i.e., the 3D hypercube) and because there is not yet any pre-trained model for these kinds of images. Thus, we decided on a two-step learning approach for implementing and training the proposed region proposal module.

In fact, CNN networks show flexibility in extracting robust spectral and spatial features from HSI image contents such as 1D, 2D and 3D CNN networks [129, 130, 139]. Thus, in this chapter, we investigate two CNN approaches for designing and implementing the proposed region proposal module: 2D–CNN networks (RPN-2DCNN model); and combining 3D–CNN and 2D–CNN networks (RPN-Hybrid-CNN model). The next sections provide a detailed explanation of these models.

The RPN-2D-CNN model

In deep learning, there are architectures that have been implemented and evaluated as global models such as the VGG models series [173]. For example, the VGG 16 model (VGG16) was implemented and then trained on global datasets (such as ImageNet dataset) for object recognition tasks. The model showed a robustness in terms of accuracy and availability (easy to implement). Thus, many research studies have used a VGG approach as a base block for their proposed approaches [175].

In VGG16, there are 13 convolutional layers, five max-pooling layers and three fully connected dense layers. The total number of trainable layers (excluding the

max-pooling layers) is 16 weight layers (VGG16). Figure 6.4 provides detailed specifications of the VGG series (as outlined in the original paper), including the number and type of layers, kernel sizes and the number of feature maps.

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Figure 6.4: Detailed specifications of VGG series architectures. The convolutional layer parameters are denoted as conv(kernel size)-(number of feature maps). The figure was copied and adapted from [173].

In our proposed RPN-2D-CNN model, as a first step, a customised version of the VGG16 model (architecture D as shown in Figure 6.4) [173] was implemented and adapted for handling an HSI image as input. This model was chosen here for its high performance in image classification tasks [173] and for having a design that is easy to modify and extend [173].

As shown in Figure 6.5, left, the model consists of five 2D–CNN blocks, including five 2D max-pooling layers with a pooling size of 2 and striding of $(2, 2)$, followed by two fully-connected layers and an output layer for two classes. In all 2D–CNN layers, we use a kernel size of (3×3) and striding of $(1, 1)$; Figure 6.5, left, shows specifications of each layer in the model such as the number of feature maps, kernel and pooling sizes, type of activation function and the size of striding operations. The

difference between the proposed model and the VGG16 model can be summarized as follows: (1) The number of feature maps of the first four blocks is empirically reduced by a factor of 2. (2) The last fully connected layer was removed and the number of nodes (or neurons) for the first two was also empirically set to 512 and 256, respectively. In fact, the main reasons behind these changes is to reduce both the number of trainable parameters and the effect of overfitting as our private dataset is limited.

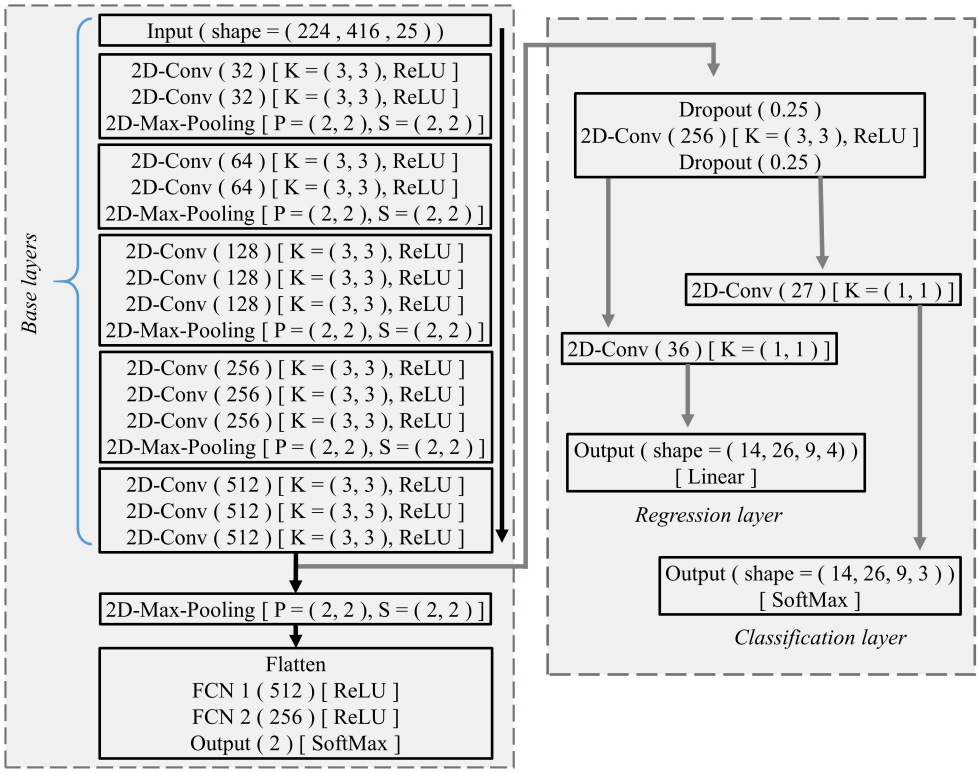


Figure 6.5: The proposed RPN-2D-CNN model for generating region proposals from snapshot HSI images. *Left*: The base-layers of the model and the whole object recognition model. *Right*: The RPN network and its architecture. K, P, S for kernel, pooling and stride size, respectively. In 2D-Conv layers, the numbers between brackets define the number of feature maps in each layer.

The main task of this model (i.e., the customised version of VGG16 model) is to recognise an HSI image (recognition model) as being normal or abnormal (i.e., with foreign materials). The recognition model was trained by using the normal and

abnormal images of our training dataset. During the training processes, we used two kinds of data augmentation processes: rotating the hypercubes with a random angle in range of $0.0 - 45$, and spatial shifting the hypercubes with random scales between $-0.2 - 0.2$. In these data augmentation processes, the affected regions in the hypercubes were filled by using the reflect filling mode. This mode defines how the input image is extended beyond its boundaries; in reflect mode, the image is extended by reflecting onto the edge of the last pixel.

In fact, the use of data augmentation aims to force the model to be invariant to local changes of objects in the abnormal images. Also, the use of the reflect mode enables the model to see images that look like totally new images, for the benefit of randomness. The model was trained for 2k epochs and it showed good convergence (i.e., an acceptable loss values) on training and validation datasets. It should be noted that after 1.5k epochs the loss on the training set was decreasing while it remained on same level for the validation set. To avoid any overfitting, training processes were stopped at 1.5K epochs, where at this point the model showed the best loss on the validation dataset.

Then, only the 2D–CNN and pooling layers of the resulting trained model, excluding the last pooling layer, were saved (for transfer learning purposes) and then used as *base-layers* for another set of 2D–CNN and as output layers for the *RPN network*; Figure 6.5, right, shows the specifications of the RPN network and how it is connected with the base-layers. It is worth mentioning that the recognition model is trained just for initializing the weights of the RPN module, and its results do not directly affect the results of the proposed RPN module as the RPN module will be trained independently (i.e., only starting the training with the weight of base-layers that were initialized with values from the first step of learning).

For the second step, the RPN network is implemented as shown in Figure 6.5, right. The architecture of the RPN network was adapted from the design of Faster-RCNN network (the RPN network) for object detection [175]. The Faster-RCNN network is chosen due its high performance in object detection tasks in global datasets such as ImageNet dataset [175].

The RPN network consists of seven layers with the following functionalities: (1) One 2D–CNN is fully-connected to the base-layers for sliding over the convolutional feature maps (i.e., the feature maps of the last 2D-CNN layer in the base-layers), here we use sliding windows of 3×3) [175]. (2) Two dropout layers were added before and after the first 2D–CNN layer to reduce the effect of overfitting in the proposed RPN module as as our private dataset is limited. (3) Two 2D–CNN layers were fully connected in parallel to the previous layers. These layers were added to compute the outputs of the RPN module (i.e., the classification output and regression output). The number of feature maps in these layers depends on the considered scales (the

used scales are 24, 48 and 64), aspect (the used aspect ratios are 1:1, 1:2 and 2:1) and classes (the classes are meat, object or background). For classification output, the number of feature maps is defined as follows: $No.\ scales \times No.\ aspect_ratios \times No.\ classes = 3 \times 3 \times 3 = 27$. Similar to regression output, the number of feature maps is defined as follows: $No.\ scales \times No.\ aspect_ratios \times No.\ coordinates = 3 \times 3 \times 4 = 36$. (4) Two reshape layers (for classification and regression) named Output layers as shown in Figure 6.5, right. These layers were added to convert the resulting output of the previous layers into a proper shape to be used as final output. Then, activation by a softmax function was applied to the output part for the classification task, while for the regression task the output was not activated (linear).

The RPN-2D-CNN model takes an HSI image as input and generates a set of feature maps (the output of the base-layers) as a 3D tensor with a size of ($14 \times 26 \times 512$) where 512 is the number of the feature maps, 14 and 26 for the width and height of the these maps, respectively. It should be noted that the width and height of the feature maps are downscaled values (due to the stride value of 2 in the pooling layers) from the input image width and height; the downscaled factor is 16 for our HSI image size. Then, these maps are fed to the RPN network through the three 2D–CNN and output layers to predict a series of candidate regions with three different scales, three different aspect ratios, corresponding classification scores and regression coefficients for each point in the spatial domain of the feature maps (i.e., 14×26).

Then, the prediction of the RPN network is postprocessed by upscaling the regression part into the original image domain. Thus, the RPN model produces a set of candidate regions (*bounding boxes*) with their classification scores for being either a background region, a meat region or a foreign object. Also, the model generates the location coefficients of these candidate boxes in the image spatial domain (i.e., x_1, y_1, x_2, and y_2 coordinates).

The RPN-Hybrid-CNN model

CNN networks are efficient models for extracting the spectral and spatial features of HSI images. These features can be jointly extracted by 3D–CNN networks. In this section, we investigate 3D–CNN network in the proposed RPN module. Thus, the RPN-Hybrid-CNN has the same structure as the RPN-2D-CNN model, where the 2D–CNN layers of the RPN-2D-CNN were replaced with 3D–CNN layers. Moreover, the number of feature maps for the first four CNN layers were empirically set to 8, 16, 32, 64 for these four layers, respectively. The RPN network for both models has the same architecture.

Thus, the proposed RPN-Hybrid-CNN model is a combination of several 3D–CNN layers, 3D max-pooling layers and 2D–CNN layers. For implementing and training the model, we follow the same approach as with the RPN-2D-CNN model (i.e., the two-step learning approach). Thus, the RPN-Hybrid-CNN model is the same as the RPN-2D-CNN model with the only difference residing in the structure of the base-layers. Figure 6.6, left, shows specifications of the base-layers in the model such as number of feature maps, kernel and pooling sizes, type of activation function and the size of striding operations. Similar to the RPN-2D-CNN model, an object recognition model, as shown in Figure 6.6, left, is implemented and trained by using the normal and abnormal images of our training dataset. Then, the weights of the base-layers are saved for further use.

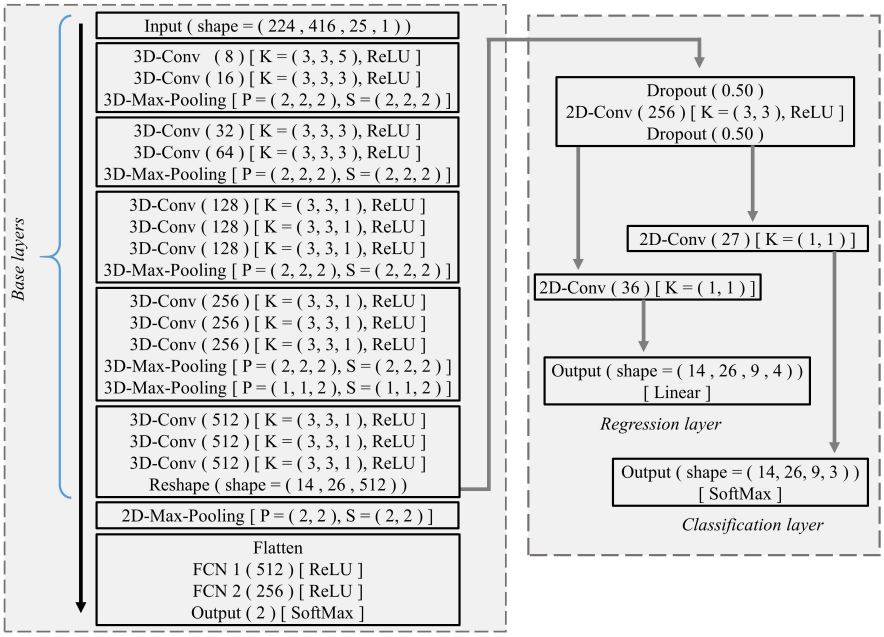


Figure 6.6: The proposed RPN-Hybrid-CNN model for generating region proposals from snapshot HSI images. *Left*: The base-layers of the model and the whole object recognition model. *Right*: The RPN network and its architecture. K, P, S for kernel, pooling and stride size, respectively. In Conv layers, the numbers in brackets define the number of feature maps in each layer.

In the base-layers of the RPN-Hybrid-CNN model, we use multiple kernel representations across the spectral dimension in the 3D–CNN layers such as $(3 \times 3 \times 5)$, $(3 \times 3 \times 3)$ or $(3 \times 3 \times 1)$. Also, the pooling and the stride operations are used for

downsampling the feature maps into a particular size. Then, for making a bridge between the 3D–CNN and 2D–CNN blocks, we add a reshape layer for having a 3D tensor of size ($14 \times 26 \times 512$). Then, the output of the reshape layer is fully-connected with the RPN network, as shown in Figure 6.6, right, similar to the RPN-2D-CNN model, where RPN networks of RPN-Hybrid-CNN and RPN-2D-CNN models are independent and similar. Thus, the output of the RPN-Hybrid-CNN is the same as that of the RPN-2D-CNN model, that is, a set of candidate regions (bounding boxes) with their classification scores and location coefficients of these candidate boxes in the image spatial domain.

Training of the RPN module

In [165,175], the RPN network was designed for generating a set of object candidates with objectiveness scores. The objectiveness score is the RPN network prediction and is used for defining the category of each candidate for being either "object or not-object". The proposed RPN models (i.e., RPN-2D-CNN and RPN-Hybrid-CNN models) are implemented individually for solving a complete classification task for each predicted candidate region. In the proposed model, each candidate has a classification score of being either meat region, foreign object or background region.

In the training phase, the same methodology is used for training both RPN-2DCNN and RPN-Hybrid-CNN models. For training the proposed models, a set of anchors needs to be generated from the input image. The anchor is a point (a position in the image) in the image domain corresponding to a point (a position in the feature maps) in the spatial domain of feature maps of the last CNN in the base-layers. The positioning from the feature maps domain is mapped onto the original image by upscaling the coordinates with a factor of 16; the factor of 16 depends on size of downsampling resulting from the pooling layers. Then, nine reference boxes are generated for each anchor by using the predefined three scales (i.e., [24, 48, 64]) and three aspect ratios (i.e., [1:1, 1:2, 2:1]). Figure 6.7 shows examples of these reference boxes for one anchor.

In fact, many of these reference boxes are not useful (not overlapped with any object in the input image) and need to be discarded in the training process. To find the best reference boxes (the useful boxes), we use IoU between these boxes and annotated bounding boxes (manually generated bounding boxes around each object in the image), then an IoU threshold could be used to define the level of confidence in selecting the best reference boxes for training purposes.

In the proposed RPN model, the classification task is to classify the proposals into into three categories: meat region, foreign object or background region. Thus, the reference boxes need to be categorised into these categories. The following IoU

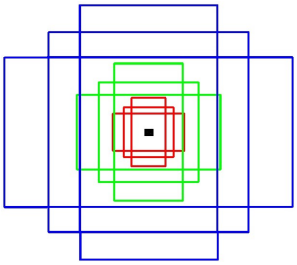


Figure 6.7: Example of the reference boxes (9 boxes) of an anchor in the image domain. The black point represents the corresponding position of the anchor in the image domain. Colours represent different scales. Changes in the size of boxes from the same colour represent the aspect ratios.

thresholds were used for selecting best reference boxes for meat regions and objects: (1) All reference boxes that have IoU with annotated bounding boxes of objects greater than 0.5 ($IoU > 0.5$) were considered as best reference boxes for the objects in the image. (2) All reference boxes that have IoU with annotated bounding boxes of objects less than 0.1 ($IoU > 0.1$) were considered as best reference boxes for meat and background regions in the image. It should be noted that the remaining reference boxes were considered as neutral boxes (not useful) and discarded from the training processes.

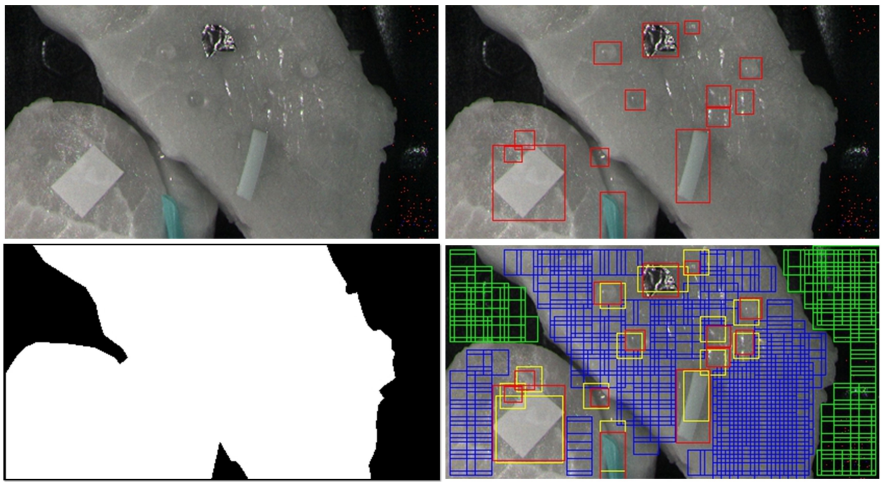


Figure 6.8: *Top*: Input snapshot HSI and its annotated bounding boxes. *Bottom*: Mask image of the input HSI image and labels of ground truth anchor boxes which are used in training the RPN models; colours Red, Green, Blue and Yellow represent ground truth of annotated objects, background anchors, meat anchors and objects anchors, respectively.

For distinguishing between the reference boxes of meat and background regions, manually generated mask images were used for selecting a set of reference boxes with a high probability of being a background or meat region. The reference boxes were cropped from the mask images, then all reference boxes with the number of zeros equal to the box size were considered as best reference boxes for background regions. A similar approach was used to find the best reference boxes for meat region (if the number of ones is equal to the box size). Figure 6.8 shows an example of the selected anchors for each class, based on the above approach for training the proposed RPN models (i.e., RPN-2D-CNN and RPN-Hybrid-CNN models).

In training processes, an adaptive mini-patch approach was used for defining the number of ground truth reference boxes of each class. In a mini-patch, the number of best reference boxes for foreign objects was selected and, randomly, the same number of reference boxes for meat and background regions was selected; this approach makes the mini-patch balanced, stochastic and representative. In back propagation processes, we used the following loss function as in [175]:

$$\mathcal{L}(\theta) = \frac{1}{N_{cls}}\mathcal{L}_{cls}(\theta) + \frac{1}{N_{reg}}\mathcal{L}_{reg}(\theta) \tag{6.1}$$

where N_{cls} is the size of the mini-patch, N_{reg} is the number of selected anchors for foreign objects, and θ represents the model parameters. The loss function, as in Eq.(6.1), is a combination of a categorical cross-entropy loss ($\mathcal{L}_{cls}$) for the classification task and the robust smooth L_1 loss [175] ($\mathcal{L}_{reg}$) for the regression task. For the regression task, the selected anchors were masked. Thus, only the anchors of foreign objects were encoded as in [175] and contributed in the loss function. For optimizing the network weights, we used the SGD algorithm for adjusting the model weights and a best model selection.

6.4.2 Filtering module

In fact, RPN networks produce much fewer candidate regions compared to a standard sliding window paradigm [165, 175]. In RPN, the number of candidates depends on the size of the feature map (14×24), not on the size of the input image as in the standard sliding window paradigm [165, 175]. There are still many candidates for the final classification and they consist of many false-alarm regions. Thus, we sought to apply a set of steps and rules, as an intermediate phase before the final classification, for filtering these candidates into those with high probability for being foreign objects. These steps are defined, as shown in Figure 6.3, as follows:

Classification: The proposed RPN network was designed for predicting a set for bounding boxes, and each box has its own probability p of being ether an object,

background or meat region in the HSI image. Thus, the rejection rule for the i-th predicted box is as follows:

$$\left\{ \begin{array}{ll} \text{Rejected}, & \text{if } \; p^i_{background} > t_1 \;\; \text{or} \;\; p^i_{meat} > t_2 \;\; \text{or} \;\; p^i_{object} < t_3 \\ \text{Accepted}, & \text{otherwise} \end{array} \right\}$$

where t_1, t_2, and t_3 are hard thresholds and set to 0.7, 0.7 and 0.8 in our experiments, respectively. By using this rule, boxes with high probability for being background and meat regions are discarded, which reduces the number of boxes by around 60% compared to the total number, and only boxes with high probabilities for being foreign objects remain for the next step.

Regression: Coordinates of remaining boxes are computed by using the predicted regression coefficient of the RPN network. Then, all boxes that have any negative coordinate (i.e., outside the image borders) or have a size less than 64 pixels are labelled as non-valid boxes (to be discarded).

Intersection: Empirically, we note that the remaining boxes (i.e., after the above classification and regression steps) are grouped as sets of boxes around each object in the image. Empirically, we observe that there are many small boxes located inside the best-predicted box. Thus, we use the intersection measure as a tool for eliminating these small boxes. First, all boxes are sorted and listed based on the probability of being objects. Second, intersections between the i-th box B_i (i.e., box with the highest probability) and the others are computed. Then, all boxes that have an intersection of 1.0 with B_i are discarded and the same process is repeated until all boxes in the list are processed.

NMS postprocessing: As in [165,175,177], NMS is used to remove the boxes that have high degrees of overlapping; IoU is used for quantifying the overlapping between predicted boxes. Empirically, we used an IoU of 0.8 as a threshold in our NMS step, with a top rule of 150 for defining the maximum number of the remaining boxes; top 150 is the maximum, while, empirically, there are fewer remaining boxes (depending on the number of objects in the image).

After applying these steps and rules, the resulting bounding boxes are ready for final classification. The resulting boxes are cropped from the input image and then resized into a fixed size of $W_s \times W_s \times 25$. These cropped boxes are used as independent HSI sub-images and as input for the next module (i.e., the classification module).

6.4.3 Classification module

Although the bounding boxes resulting from the RPN and filtering modules are very good candidates for FOD, still some boxes are related to background or meat regions in the image. For an accurate FOD, we add a classification module for classifying the resulting bounding boxes (i.e., after applying region proposal and filtering modules) into background, meat or foreign objects by using CNN networks.

CNN networks are an efficient tool for using both spectral and spatial information of HSI images. Instead of classifying bounding boxes based on the feature maps of the RPN model as in[165, 175], we propose applying an independent CNN classification model, by 3D–CNN networks, for final prediction in the proposed FOD model. Thus, we enforce the FOD model for better utilizing and joining both spectral and spatial features of HSI images.

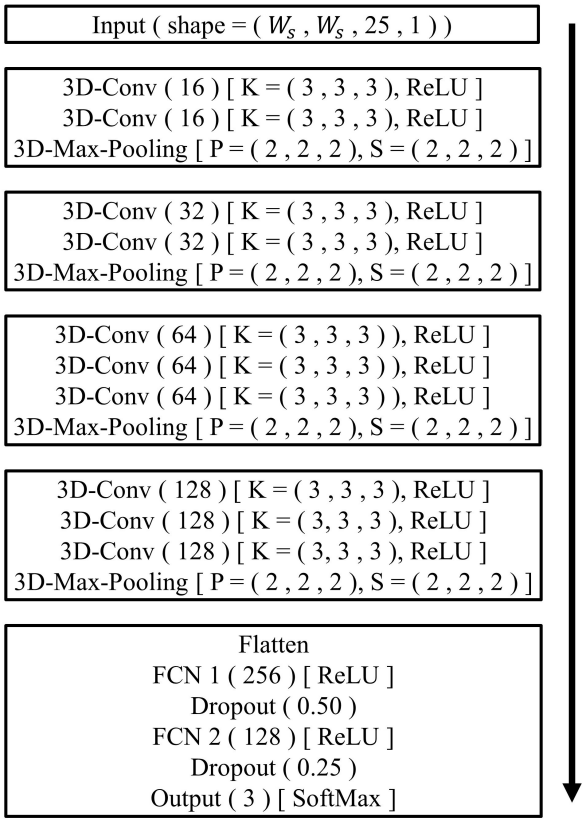


Figure 6.9: The proposed 3D–CNN architecture for final classification task in the proposed FOD model.

3D–CNN model

The proposed 3D–CNN model, as shown in Figure 6.9, consists of a hierarchical structure of ten 3D–CNNs with a kernel size of $3 \times 3 \times 3$, four 3D max-pooling layers with a pooling size of 2 across each dimension, two dropout layers with a ratio of 0.5 and 0.25 and two fully-connected layers of size 256 and 128. For classification, a softmax layer (with 3 nodes) was added as an output layer, where 3 defines the predefined classes (i.e., background, meat and foreign object). In all 3D–CNN and fully-connected layers, we use ReLU functions for activating the output of these layers. Figure 6.9 shows the specifications of the proposed 3D–CNN classification architecture.

Training the classification module

For training the proposed classification model (i.e., the proposed 3D–CNN), the same training approach was applied based on the defined training dataset of HSI images. Iteratively, a random HSI image is selected from the training dataset. Then, the reference anchors of the image are computed based on the defined scales and aspect ratios, which means that the model was trained on the anchor space not on the predicted bounding boxes of the region proposal module.

Next, the anchors are labelled, for having their ground truth, by using the IoU measure and the masked images. Besides the annotated bounding boxes of objects, the anchors with $IoU >= 0.5$ are labelled as objects, while the anchors with $0.3 > IoU > 0.5$ are considered as neutral and they are ignored during the training processes. The remaining anchors are labelled into meat and background regions based on the predefined masked image of the input HSI image.

Adaptively, a set of random anchors is selected from each class and cropped into a fixed size of ($W_s \times W_s \times 25$); the patch size is adaptive and depends on the number of best anchors that have high overlapping with the annotated ground truth. The selected patch is now fed into the model and a categorical cross-entropy loss function, as defined in Eq. (4.16), is used for computing the error between actual (i.e., the ground truth) and predicted outputs, while an SGD algorithm is used for optimising the model weights.

In the testing phase, the resulting candidate boxes by the region proposal module are fed to the classification module for having the final probability of being either a foreign object, meat or background region. Then, a hard threshold (t_4) is used for keeping only boxes with high probability of being foreign objects and eliminating the other candidates; empirically, t_4 was set to be 0.8 based on our experiments and datasets.

6.5 Selective search for region proposal

Selective search is a region proposal approach used in localization of objects in an image. It is proposed to enhance the accuracy and speed of a sliding window approach. The algorithm selective search aims to design a fast algorithm with a very high recall rate ensuring that all objects are detected. A hierarchical grouping approach is applied to merge similar regions (i.e., proposals) based on a set of properties such as colour, texture, size and shape compatibility [178].

First, the algorithm starts with segmenting (over-segmenting) the input image by using by a graph-based segmentation based on intensity of the pixels. It should be noted that prefect segmentation is not the goal of the algorithm, where the resulting segments are utilized as initial seeds of the proposals. Second, the proposals are grouped based on the following similarity metrics:

Colour similarity by calculating the histogram (25 bins) of each channel for each proposal box, obtaining a 75-dimensional colour descriptor for each proposal. Then, the colour similarity is computed as follows:

$$S_{colour}(p_i, p_j) = \sum_{k=1}^{n} \min(c_i^k, c_j^k) \tag{6.2}$$

where c_i^k is the k-th value in colour descriptor of the proposal p_i.

Texture similarity by calculating Gaussian derivatives at 8 different orientations for each channel in a proposal. Then, a histogram of 10-bins is calculated for each orientation in each channel, obtaining 240-dimensional texture feature descriptor for each proposal. Then, the texture similarity is computed as follows:

$$S_{texture}(p_i, p_j) = \sum_{k=1}^{n} \min(t_i^k, t_j^k) \tag{6.3}$$

where t_i^k is the k-th value in colour descriptor of the proposal p_i.

Size similarity for merging regions that have small sizes. The size similarity is computed as follows:

$$S_{size}(p_i, p_j) = 1 - \frac{size(p_i) + size(p_j)}{size(img)} \tag{6.4}$$

where $size(img)$ is size of the input image in pixels.

Shape compatibility for measuring the fitness between two regions and how they fit into each other. The shape compatibility is computed as follows:

$$S_{shape}(p_i, p_j) = 1 - \frac{size(bb_{ij}) - size(p_i) - size(p_j)}{size(img)} \tag{6.5}$$

where $size(bb_{ij}$ is a bounding box around both proposals p_i, p_j.

Finally, these similarity metrics are combined (linear combination) together to form the final metric that is used in the algorithm. Thus, the final similarity metric between any two proposals is defined as follows:

$$S_{final}(p_i, p_j) = w_1 * S_{colour} + w_2 * S_{texture} + w_3 * S_{size} + w_4 * S_{shape} \tag{6.6}$$

where w values ($w \in 0, 1$) are weights defining the contribution of each similarity metric to the final metric. The resulting final segments, after these merging processing, are used to extract the best fitted bounding boxes around each segment. These bounding boxes are the final output of the selective search algorithm.

The selective search algorithm, as proposed in [178], was designed for handling RGB digital images. For HSI images, the algorithm needs to be adapted. Thus, both colour and texture similarities need be adapted due to the dimensionality of HSI images. To match with these similarities, we applied PCA to reduce the dimensions of the image. Then, the first three PCA components were extracted for each image in the dataset. To compute the similarities as defined in Eq.(6.2) and (6.3), the three colour channels were replaced with the three PCA components of the HSI image.

6.6 Experimental results and analysis

In training of both region proposal and classification models, we used an SGD algorithm with a momentum of 0.9, a weight decay of 0.0001 and an initial learning rate of 0.001. The models were trained for $5K$ epochs; after each $1K$ epochs, the learning rate decayed by 0.1%. The spatial window size (W_s), in the classification module, was empirically chosen as (64×64). Thus, the cropped windows have a size of ($64 \times 64 \times 25$).

The localisation accuracy of the candidate regions is very important for having an accurate object detection algorithm [165,175]. Thus, we first evaluated the performance of the proposed region proposal models for both the validation and testing

dataset. For comparison, the common selective search method [50,165,174,175,178] was implemented for evaluating the robustness of the proposed RPN models. For evaluating the proposed RPN models and the selective search method, we used *recall rate* (RR) evaluation, which is defined as follows:

$$RR = \frac{\textit{Number of correct predictions}}{\textit{Total number of objects}} \tag{6.7}$$

where the number of correct predictions (or recall number) is computed based on the IoU between the prediction and the ground truth. The total number of objects is fixed for each dataset and equal to 1391 and 2153 for the validation and testing datasets, respectively.

Table 6.1 shows the performance of the RPN models in comparison with the selective search method. The RPN models achieve a higher RR for different IoU thresholds, including 0.5, 0.4 and 0.3, compared to the selective search method. On average for the IoU thresholds, the RPN model by hybrid-CNN approach provides an RR of 86.7% and 95.1% for the validation and testing datasets, respectively, while the RPN model by 2D–CNN approach achieves RR of 73.6% and 97.0%. In both approaches, the accuracy of models on the testing dataset is very close, while the hybrid approach outperforms the 2D–CNN approach in the validation set of images.

The selective search method achieves low RR, compared with the proposed RPN models, of 58.8% and 91.0% for validation and testing datasets, respectively. Moreover, both RPN models show accurate localisation (i.e., a high RR with an IoU of 0.5) and detection (i.e., a high RR on average of three IoUs), with fewer candidates (i.e., top 150 rule) in comparison to the selective search method. These results show that RPN models are efficient tools for having accurate candidates of foreign objects in an HSI image.

The results, as shown in Table 6.1, showed that the RR of the testing dataset is higher than the RR of the validation dataset. In fact, in testing a set of images, a totally new set of materials were added (as foreign objects) to a totally new set of meat samples. In these foreign objects, we do not include any glass or transparent materials. While in the validation dataset, many glass and transparent materials were included in the experiments as foreign objects. Thus, the difference in RR values of both datasets could be interpreted as a kind of failure in the model in detecting these kind of materials (glass and transparent materials).

For visual comparison, Figure 6.10 shows the resulting candidate region, by using each method, on selected images from the validation and testing datasets. Clearly, the results show that the proposed RPN models produce candidates clustering around the foreign objects in the images. Thus, candidate regions of the RPN models look accurate and are distributed only around the foreign objects in the image, while normal regions in the image look "clean from detection".

Table 6.1: Performance evaluation on HSI images for the proposed RPN models, in comparison with the standard selective search method. "Avg No" shows the number of generated regions, on average for the whole dataset, by each method.

Method	**IoU**	**Validation set**			**Testing set**		
	$>$	Recall No.	RR	Avg No.	Recall No.	RR	Avg No.
Selective search	0.5	658	47.3	183	1793	83.3	212
	0.4	834	59.9		1993	92.5	
	0.3	960	69.1		2090	97.1	
	avg		58.8			91.0	
RPN (2D–CNN)	0.5	835	0.60	Top50	1934	88.7	Top50
	0.4	883	63.1		1968	90.9	
	0.3	909	65.2		1980	92.1	
	avg		62.8			91.2	
	0.5	925	66.3	Top100	2025	94.2	Top100
	0.4	977	70.5		2067	96.3	
	0.3	1020	72.8		2074	95.7	
	avg		70.1			95.2	
	0.5	963	69.2	Top150	2064	96.1	Top150
	0.4	1029	74.3		2098	97.2	
	0.3	1080	77.9		2107	98.1	
	avg		73.6			97.0	
RPN (Hybrid-CNN)	0.5	998	71.7	Top50	1885	87.6	Top50
	0.4	1080	77.6		1943	90.2	
	0.3	1130	81.2		1976	91.8	
	avg		76.9			89.9	
	0.5	1091	78.4	Top100	1961	91.1	Top100
	0.4	1191	85.6		2023	93.9	
	0.3	1244	89.4		2052	95.3	
	avg		84.5			93.5	
	0.5	1118	80.4	Top150	1995	92.7	Top150
	0.4	1222	87.9		2057	95.5	
	0.3	1276	91.7		2088	97.1	
	avg		86.7			95.1	

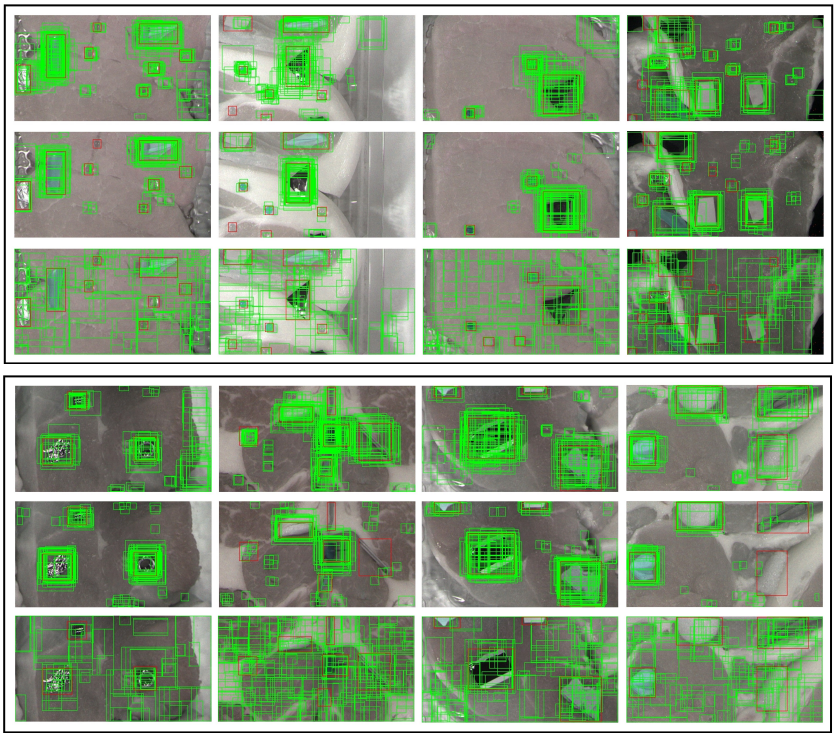


Figure 6.10: Visual comparison between resulting candidate boxes of the 2D–CNN RPN network (*top rows*), Hybrid-CNN RPN network (*middle rows*) and selective search (*bottom rows*). Top and bottom groups are selected images from validation and testing sets, respectively.

For evaluating the proposed FOD framework, we provide comparisons with an FOD model by selective search method. *Average precision* (AP) score is a popular evaluation metric in measuring the accuracy of object detection models [165, 175, 176, 179, 180]. AP computes the average precision value for recall value over 0 to 1. Thus, AP takes the proportion of true-positive (precision) and the proportion of true-positive out of the possible positives (recall) into consideration.

Practically, the AP score for a dataset is computed as in the following steps: (1) The prediction of the model is collected for objects in all images and ranked in descending order according to the predicted confidence scores (the final output of the classification model). Then, the predictions are classified into correct or not based on an IoU threshold. (2) For each prediction, we compute the recall and precision scores. (3) We compute the AP score of the model prediction as the area under

precision-recall curve. Thus, AP is the interpolated area under the curve and it is computed as follows:

$$AP = \frac{1}{11} \sum_{Recall_i} Precision(Recall_i) \tag{6.8}$$

where $Recall_i = [0, 0.1, 0.2, ., ., 1.0]$ are 11 equally spaced recall values that are used in the interpolation process.

For FOD performance evaluation, we use AP, following [165, 175, 176, 179, 180], as a measure based on three IoU thresholds [0.5, 0.4, 0.3]. For fair comparison, the same classification model (i.e., the proposed 3D–CNN model) is used as the final classifier for all FOD models. It should be noted that the AP sore is computed for the whole dataset not for each image individually.

Table 6.2 shows the evaluation of the proposed FOD models, and FOD by selective search. The resulting AP shows that FOD with RPN outperforms FOD with the selective search method, both on the validation and testing set of HSI images. On average of the three IoUs, the FOD by selective search achieved very low APs of 39.9% and 50.6% for validation and testing, respectively.

Table 6.2: Performance evaluation of the proposed FOD models for HSI images in comparison with a model following the selective search method.

FOD model		**Validation set**				**Testing set**			
Region proposal method	Classification model	**AP@**			**avg**	**AP@**			**avg**
		0.5	**0.4**	**0.3**		**0.5**	**0.4**	**0.3**	
Selective search	3D–CNN	26.2	40.3	53.1	39.9	31.7	52.2	67.9	50.6
RPN (2D–CNN)	3D–CNN	57.2	61.6	63.7	60.8	75.9	82.1	85.1	81.0
RPN (Hybrid–CNN)	3D–CNN	59.9	64.9	66.9	63.9	70.6	76.8	79.9	75.8

The FOD models by RPN approach provided a much higher AP, on average, for both datasets. The FOD model by RPN-2D-CNN networks outperformed the FOD model by RPN-Hybrid-CNN networks on the testing dataset with APs of 81.0% and 75.8%, while vice versa for the validation dataset with APs of 60.8% and 63.9%. This variation in results shows that the accuracy of the FOD model is exactly following

the accuracy of the region proposal method. Thus, the RPN network is more accurate both for FOD localisation and detection.

By averaging the results over the two datasets, the FOD model by RPN-2D-CNN network is more accurate, on both datasets, both for FOD localisation and detection with AP, on average of the three IoUs, of 81.0% and 60.3% for testing and validation, respectively. Thus, we suggest the FOD model by RPN-2D-CNN network as the best model based on our experiments.

Regarding object detection problems, visual representations are very important for visually showing the complexity of the problem (the challenge), the accuracy of prediction and the localization of predicted objects. Figures 6.11 and 6.12 show the final output of the proposed best FOD model (i.e., the selected best FOD by RPN2D-CNN network and the 3D–CNN model) on a selected set of snapshot HSI images from the validation and testing datasets, respectively.

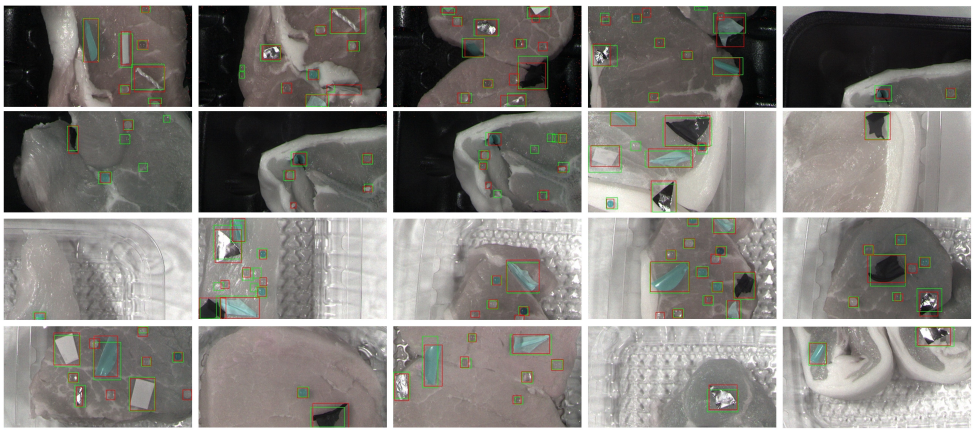


Figure 6.11: Visual results of the best proposed FOD model on selected HSI images from the validation set showing a collection of meat samples (lamb, beef and pork). Red and Green for ground truth and predicted bounding boxes, respectively.

The visual representations clearly show that the proposed FOD model is able to understand the normal regions (i.e., meat and background regions) in the images, where these regions accurately look clean from detection. Also, with the large scale of considered objects, the foreign objects (the abnormal regions) were accurately detected despite their differences in shape, colour, size, types and locations. However, the presence of objects like glass or other transparent materials decreased the accuracy of detection.

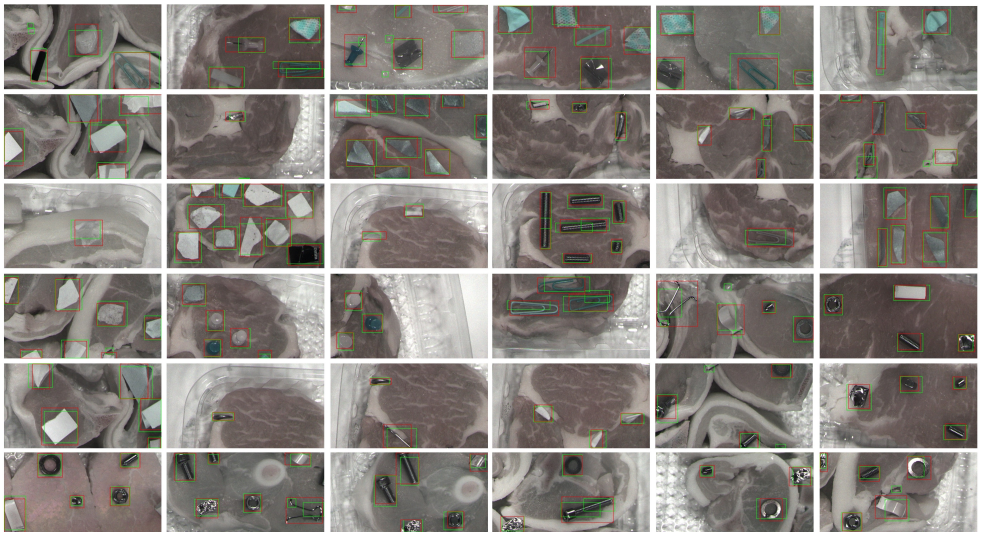


Figure 6.12: Visual results of the best proposed FOD model on selected HSI images from the testing set showing a collection of meat samples (lamb, beef and pork). Red and Green for ground truth and predicted bounding boxes, respectively.

Table 6.3: Performance evaluation of the proposed best FOD model on each meat type from the validation set.

Meat type	**AP@**			**avg**
	0.5	0.4	0.3	
Lamb	45.5	50	52.3	49.3
Beef	65.8	69.6	71.6	69.0
Pork	52.4	57.1	59.3	56.3

The performance of the proposed best FOD model (i.e., the FOD by RPN-2D-CNN network and the 3D–CNN model) was further investigated regarding the effect of each meat type on the performance. Table 6.3 provides achieved AP results of the model on different meat products, including lamb, beef and pork products. The model provided AP (on average of the three IoU thresholds) of 49.3%, 69.0% and 56.3% for lamb, beef and pork, respectively.

Thus, this variation in performance among the meat types suggests an impact of meat texture on the prediction model, where the texture of beef has the lowest impact and that of lamb has the highest. The effects of meat may also be associated with the amount of specular reflectance due to the presence of water on the surface of meat samples.

Figure 6.13 shows visually the impact of specularities on model prediction for different meat types. These specular regions appear as false detections in lamb and pork images, while beef images appear to be free from such regions, thus with less falsely detected objects. Overall, these results suggest that the proposed FOD is using both the spectral and spatial features of HSI images in an efficient way.

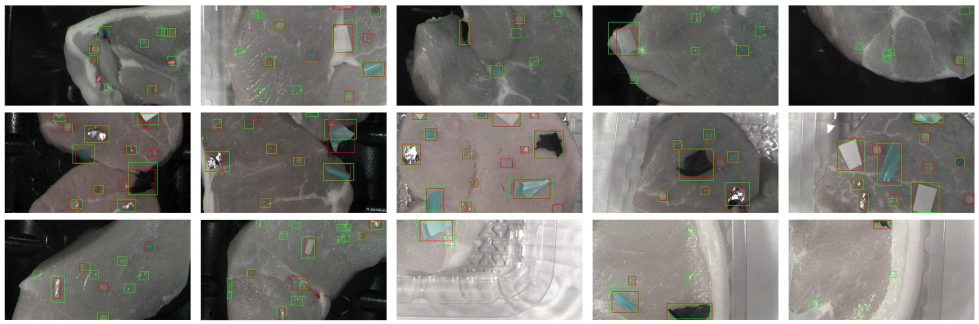


Figure 6.13: Visual results of the best proposed FOD model on selected HSI images from each meat type. Rows from *top to bottom* are the model prediction on lamb, beef and pork images, respectively. Red and Green for ground truth and predicted bounding boxes, respectively.

For showing the robustness of the proposed model by HSI imaging against standard RGB imaging, we implemented the following simple experiment: RGB images of meat samples were collected and then printed by a colour printer, then the printed images were used as foreign objects to contaminate the same samples (to visually look like the meat or fat regions in the scene), then the samples were imaged by both RGB and the snapshot HSI cameras; Figure 6.14, top, shows examples of these RGB images with the foreign objects. In Figure 6.14, top, and qualitatively, we can note that any object detection model, by using RGB imaging, could fail in this task, whereby it is difficult with the human eye to recognise or localise these foreign objects.

By qualitative evaluation, the proposed FOD provided an accurate detection and localization of these foreign objects as shown in Figure 6.14, bottom. Thus, this simple experiment shows that the proposed FOD model is fully utilizing both spectral and spatial features of HSI images in an efficient way. Moreover, this experiment jus-

tifies the use of snapshot HSI in research and for solving the problem (i.e., foreign objects detection in meat products), where HSI shows an efficiency for materials discrimination based on their chemical and visual properties.

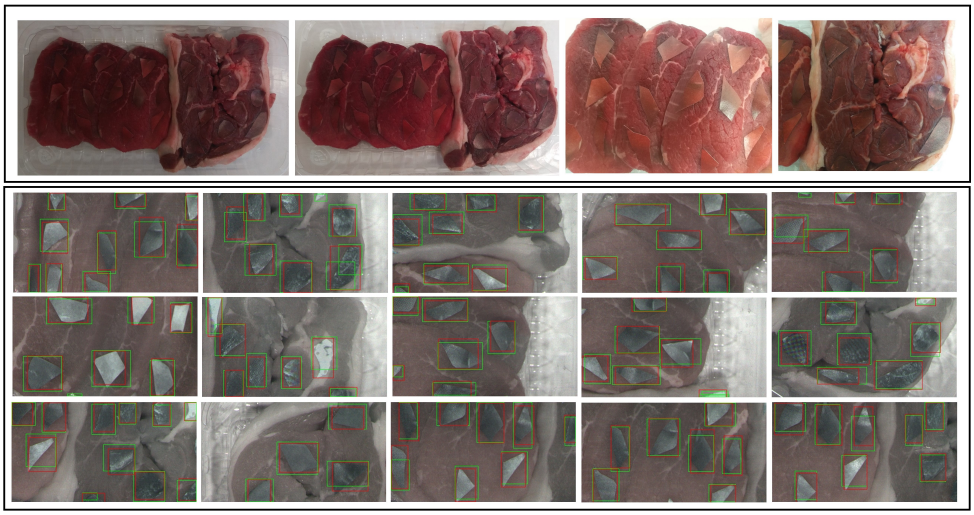


Figure 6.14: Visual results for showing the robustness of the proposed model by HSI imaging against the standard RGB imaging. *Top*: RGB images of meat samples are contaminated with foreign objects with the same visual properties as the meat samples. *Bottom*: Prediction results of the proposed FOD model on HSI images of the same samples as in the top. Colours, Red and Green for ground truth and predicted bounding boxes, respectively.

We also analysed the run-time of the investigated models and methods; all methods and models are implemented on the same machine. Results show another efficiency of the RPN network; run-times (on average for all images in the datasets) are 0.06 sec and 1.5 sec for RPN and selective search, respectively. Run-times of complete FOD frameworks are 1.3 sec and 4.6 sec with RPN and selective search, respectively. These results show that RPN is much faster than selective search for FOD, which defines the RPN model as being more relevant for real-time applications in the meat industry.

From an actual production perspective and application point of view, we further analysed the prediction of the proposed FOD model regarding the efficiency of the model in detecting all foreign objects in the image and ensuring that the model is not detecting any normal regions (i.e., meat or background) in the image as foreign objects. To quantify this analysis, we use *false negative rate* (FNR) and *false positive*

rate (FPR) as evaluation metrics. False negative rate (also called *miss rate*) measures the proportion of false negatives in the prediction, where a very low value (FNR = 0) means perfect prediction. Similarly, false positive rate (also called *fall-out rate*) measures the proportion of false positives in the prediction. Practically, FNR and FPR are computed as follows:

$$FNR = \frac{FN}{FN + TP} \tag{6.9}$$

$$FPR = \frac{FP}{FP + TN} \tag{6.10}$$

where FN, TP, FP, and TN are the false negative, true positive, false positive and true negative values of the confusion matrix.

To compute the confusion matrix, the ground truth and IoU thresholds were used to define the FN, TP, FP and TN values. For example, if the IoU between the ground truth box and the predicted box is above the threshold, the predicted box was considered as TP.

Table 6.4: Proportion of false positives and false negatives in the prediction of the proposed FOD model on each meat type from the validation set and on the testing dataset with mixing all meat types in images.

		FPR						**FNR**					
Data set	**Meat**	**IoU**					**avg**	**IoU**					**avg**
		0.5	0.4	0.3	0.2	0.1		0.5	0.4	0.3	0.2	0.1	
Validation	Lamb	0.39	0.36	0.35	0.35	0.34	0.35	0.42	0.33	0.31	0.30	0.30	0.33
	Beef	0.18	0.15	0.14	0.13	0.12	0.14	0.31	0.26	0.23	0.22	0.21	0.24
	Pork	0.32	0.30	0.29	0.28	0.27	0.29	0.33	0.29	0.26	0.24	0.23	0.27
	avg	0.29	0.27	0.26	0.25	0.24	0.26	0.35	0.29	0.26	0.25	0.24	0.28
Testing		0.27	0.24	0.21	0.18	0.17	0.21	0.18	0.12	0.10	0.08	0.08	0.11

Table 6.4 shows the achieved FNR and FPR results for the validation and testing datasets. The results show a promising performance of the proposed FOD model with good and acceptable FNR and FPR rates. In the validation dataset, the analysis was performed for each type of meat to show the effect of meat type in the model prediction. In case of images with beef meat, the model provided the lowest FPR rate (FPR = 0.14 on average of all IoU thresholds) and lowest FNR rate (FNR = 0.24 on average of all IoU thresholds). On average of the validation set, the model achieves 0.26 and 0.28 for FPR and FNR, respectively. These results show that the model has the same level of understanding for both normal and abnormal regions in the image.

The results for the testing dataset, as shown in Table 6.4, show an enhancement of FPR and FNR values compared with the validation set; the testing dataset consists of a new set of meat samples and a new set of foreign objects. The model achieved FPR rate is 0.17 on average of all IoU thresholds, which represents the number of foreign objects that could fail in the meat products. Moreover, the model shows good results in detecting only foreign objects, that is, not detecting any region from meat or background (FNR = 0.11 on average of all IoU thresholds).

As practical aspects, the model still has the disadvantage of rejecting or accepting some products that should not be rejected or accepted. This observation is due to the achieved FPR and FNR values, where these are supposed to be very close to zero for the perfect model. This disadvantage of the model could be improved by considering multiple types of models at different stages of inspection. Moreover, this limitation in the proposed model defines a good research opportunity for further research into calibrating the proposed approach for a particular inspection problem in the meat industry.

6.7 Summary

This chapter shows that the use of HSI images for FOD in food products is an important opportunity for the food industry. The proposed framework consists of three modules in sequential order: an RPN, a filtering module and a classification module. The proposed FOD framework was designed for using both the spectral and spatial features of HSI images. The proposed framework was applied and evaluated on red-meat products as a case study, while the same approach could be fitted on other applications in the food and agriculture industries.

Quantitative and qualitative results show that the proposed FOD framework outperforms the state-of-the-art methods in terms of accuracy of detection, localisation of detected objects and computation time. Moreover, the results show that

the proposed FOD framework is efficient by using and understanding both spectral and spatial features of HSI images. The proposed FOD framework is an efficient solution for completely portable HSI systems (mobile HSI systems) and FOD of video HSI by using snapshot HSI technology.

Chapter 7

Conclusions and Future Work

The main goal of using hyperspectral imaging systems (HSI) is to obtain chemical (spectral) and spatial distributions of materials in the meaning of an image. These distributions help to deepen understanding of materials in many computer vision tasks, including segmentation, classification and object detection. Thus, HSI imaging has received more attention in research regarding various applications: remote sensing, food processing and medical applications. In the food industry, prediction models based on HSI image data have been widely used for estimating a broad range of attributes related to the safety and quality of food products.

Motivated by advancements in the analysis of interactions between spectral and spatial distributions of HSI images, by the new snapshot HSI sensors, and by the recent revolution of deep learning in computer vision research, this thesis contributes to developing advanced and novel deep learning models for solving several research problems in the meat industry: the red-meat adulteration problem; the red-meat authenticity problem as fine-grain materials classification; and foreign object detection in meat products. The developed models may contribute in the general food industry towards more accurate, portable and real-time applications by using snapshot HSI sensors.

This chapter reviews and summarizes the main findings of this thesis and provides directions for proposed future work related to the subjects of this thesis.

7.1 Conclusions

The general objective of this thesis is to develop advanced spectral-spatial processing techniques for effectively addressing the interaction between spatial and spectral distributions of materials which have been made accessible by recent growth in HSI systems.

This thesis includes various approaches for HSI data collection and preparation, feature extraction, HSI data classification and object detection and localization in

HSI images as presented in Chapters 2, 4, 5 and 6. The main findings of the thesis are summarized as follows:

- Chapter 2 presents the practical and technical fundamentals of HSI imaging technology, including HSI image structures and notations, the methods for HSI image acquisition and generation and the approaches for sensing HSI data. Two HSI imaging systems were implemented, in reflectance sensing mode, and then used to collect our private datasets: line-scanning and snapshot HSI systems. For reproducibility purposes, all details of these systems are provided in this chapter such as the used equipment, system parameters and optimization methods, and reflectance and illumination correction methods. The main result of this chapter is the establishment of two HSI systems that were used for collecting high-quality HSI images by the standard line-scanning HSI system and the portable and high-speed snapshot HSI system. Moreover, the experimental setup of these HSI systems could support related research in the collection of HSI images, in the indoor environment, for other kinds of applications.

- Chapter 4 provides a comprehensive analysis and comparison by investigating and proposing several approaches for HSI feature extraction in HSI classification tasks: spectral features; handcrafted spectral and textural features; and self-extracted features. These features were evaluated on the red-meat adulteration detection problem by a private dataset collected by the implemented line-scanning HSI system. In the case of self-extracted features, a novel multi-structure CNN (deep learning) model is proposed by combining two CNN network structures in a single model: 1D and 3D CNN networks. The comprehensive results showed that the proposed approaches and models outperform the state-of-the-art approaches and models by both qualitative and quantitative evaluation. Also, the experimental results showed that considering the interaction between spectral and spatial features of an HSI image plays an important role in enhancing the accuracy of an HSI classification model. Moreover, the proposed CNN model shows robustness, for red-meat adulteration detection, in terms of the overall accuracy of 94.4%, independent of the status of the meat (fresh, frozen, or thawed) and the ability to handle raw HSI data without any preprocessing or presegmentation methods. The proposed CNN model was demonstrated to be more suitable for handling chemical composition and textural distributions of red-meat species, which suggests that the model can be applied for other applications in the food industry.

- In Chapter 5, the potential of new HSI snapshot systems were investigated for the red-meat authenticity (or red-meat classification) problem as a case study

of fine-grain materials classification tasks. Moreover, in this chapter, joining both spectral and spatial features of an HSI image is addressed; the joint features were extracted by using single operation approaches such as 3D–CNN operations across a spatial size of the whole HSI cube. The proposed 3D–CNN model was evaluated on private datasets of HSI images of red-meat products from three HSI systems: line-scanning, NIR snapshot and VIS snapshot systems. The quantitative and qualitative results showed that the proposed model achieved excellent classification accuracy on images of these three HSI systems, also the model outperformed the state-the-art models on the data of the three systems. Moreover, a novel graph-based postprocessing method is proposed in this chapter for enhancing the prediction of any deep learning model for pixel-wise classification. The evaluation of the method showed that the proposed method provides significant enhancements on the prediction of the 3D–CNN model without losing any information from the classification maps such as maintaining the edges between the classes. Also in this chapter, we provided benchmark results as a comprehensive comparison between the three HSI systems on the same research problem. The comparison reported the following research observations: the need for spatial features (or the joint features) is dependent on the available spectral information (inverse relationship); the accuracy of the shallow machine learning models is also dependent on the available spectral information (direct relationship); and the deep learning model, by joining spectral and spatial features, is fairly stable and provides a robustness for utilizing all available HSI image data. Finally, despite the limitation in spectra of snapshot HSI images, the proposed model and approach by the joint features provide an accurate classification of all classes, which suggests that the model is robust for real-time food processing applications in the food industry, where these snapshot systems are portable and able to work at the video rate.

- In Chapter 6, a novel deep learning framework for object detection in HSI images is proposed for the first time. As a case study, we used FOD in meat products for evaluating the proposed framework; two independent datasets of NIR snapshot HSI images of meat containing a large scale of foreign materials were used in the evaluation processes. The proposed framework has a nature of sequential processes of three main modules: RPN for generating a set of candidates with probabilities of being as objects; filtering for enhancing the resulting candidates by certain rules; and classification for classifying the remaining candidates, based on their spectral and spatial features, into object, meat or background. The quantitative results showed that the proposed FOD model provides much high average precision, in both datasets, in comparison with

an FOD model following the selective search approach. Moreover, the quantitative analysis showed that the proposed FOD provides accurate results for detecting and localising foreign materials in meat products, independently of their shapes, sizes and colours, which demonstrates that the proposed model is fully utilizing the spectral and spatial features of the snapshot HSI images. Also, the proposed FOD model showed efficiency in run-time in comparison with the baseline model. Thus, this study (i.e., the study conducted in Chapter 6) shows opportunities for real-time materials object detection based on joint spectral and spatial features of these materials and opens doors for real-time object detection applications in the food industry, where the achieved run-time (1.3 sec/image) is a competitive result for fulfilling the real-time requirement of these kind of applications (i.e., HSI for food processing).

7.2 Future work

This thesis focus is on implementing advanced techniques for better utilizing spectral and spatial features of HSI images by using advanced deep learning models. Thus, as future work, we would be interested in investigating open research problems in this young research area. Suggested future directions are as follows:

- First, in Chapters 4 and 5, we used local classification operations to classify meat images (i.e., the pixel-wise classification approach). In this approach, the models were forced to learn local features of meat samples. Although we achieved accuracy, the models could define a drawback for run-time in the testing phase. The existing deep learning models for semantic segmentation can accept the whole image as input and provide output as labels for each pixel in the image; these models mostly learn global features from the whole image such as the shape of meat samples or the distribution of sample content. Thus, adapting these models (i.e., semantic segmentation models) to learn local spectral and spatial features of meat or any other kind of material defines a good area of research with the benefit of having accurate models and the possibility of reducing the run-time of proposed models.

  Moreover, the raised run-time issues in these chapters could be improved in future by implementing the proposed pixel-wise classification framework using parallel computation approaches, where the classification processes of the pixels are independent processes. This suggests that parallelization of these processes could be implemented on both software and hardware levels.

- Second, in Chapter 6, we used the sequential and multi-step approach for better utilization of spectral and spatial features of HSI images for solving an object detection task. The proposed model can be used, in future, for other applications such as microbial defect detection in meat or other food products, fruit sorting and object tracking by using their spectra (e.g., car tracking in driving assistant applications). Moreover, the proposed models can be improved, in terms of accuracy and run-time, by sharing the operations of RPN and classification steps in an efficient way to enhance the use of spectral and spatial features of input images.

Appendix A

Supplementary Materials

This appendix provides supplementary materials as complementary quantitative and qualitative results for the research reported in this thesis. We feel that these results might be of interest for supporting further the achieved results and observations in this thesis.

The appendix is organised as follows. Section A.1 provides additional quantitative and qualitative results regarding the research in Chapter 4. Section A.2 shows more quantitative results for Chapter 5. Finally, Section A.3 presents visualization results of the proposed FOD model that is proposed in Chapter 6.

A.1 Supplementary material for chapter 4

In Chapter 4, we provide a comprehensive analysis of utilizing the only spectral features for classifying HSI image data by using SVM and PLS–DA models. In this section, we provide detailed results (quantitative and qualitative) of these models for each considered type of spectral features and for each meat status.

Table A.1 shows results of both SVM and PLS–DA models when only spectral features were taken into consideration. F_1 measure and overall accuracy were computed for each model by the following feature vectors: raw spectral, 1st derivative spectra, 2nd derivative spectra, L_2 normalization, and SNV normalization. Also, evaluations are provided for each meat status (i.e. FSUP, FSP, FRUP, FRP, and THUP).

Figure A.1 provides a classification map for the SVM model for the following feature vectors: raw spectral, L_2 normalization, L_2 normalization with texture, SNV normalization, SNV normalization with texture. Also, classification maps for each meat status are available in Fig. A.1

Table A.1: Performance evaluations of both SVM and PLS–DA models on different spectral feature vectors at each meat status.

Model	Meat status	Feature vector	F_1 score			O.A
			LAMB	OTHER	FAT	
SVM	FSUP	Raw spectral	88.2	93.5	97.0	92.8
		1st derivative	88.5	92.1	97.5	93.1
		2nd derivative	87.1	90.5	97.1	92.8
		L_2-norm	87.5	92.6	99.4	92.6
		SNV-norm	94.2	96.2	98.0	96.1
	FSP	Raw spectral	74.8	82.0	96.2	83.3
		1st derivative	77.1	83.1	97.1	85.3
		2nd derivative	74.2	80.1	96.8	83.1
		L_2-norm	72.1	79.1	98.6	81.5
		SNV-norm	76.1	82.7	97.0	84.1
	FRUP	Raw spectral	78.5	85.1	98.2	86.3
		1st derivative	80.1	87.2	97.9	87.1
		2nd derivative	78.2	80.1	96.8	86.7
		L_2-norm	82.0	87.2	97.1	88.1
		SNV-norm	89.8	93.8	98.3	93.9
	FRP	Raw spectral	80.2	86.7	95.3	87.1
		1st derivative	82.5	87.5	97.8	89.1
		2nd derivative	80.7	84.2	97.1	87.1
		L_2-norm	83.5	89.5	95.9	89.5
		SNV-norm	86.7	92.5	96.9	92.1
	THUP	Raw spectral	79.6	86.3	96.3	86.8
		1st derivative	80.5	86.5	96.8	88.5
		2nd derivative	79.2	83.6	95.5	86.6
		L_2-norm	75.7	82.8	98.9	84.4
		SNV-norm	85.1	90.5	97.7	90.7

PLS–DA	**FSUP**	Raw spectral	72.2	82.5	79.3	77.7
		1st derivative	70.7	81.6	77.1	76.3
		2nd derivative	71.4	83.9	78.3	77.7
		L_2-norm	73.2	83.4	88.1	81.4
		SNV-norm	74.5	84.2	87.0	82.0
	FSP	Raw spectral	65.6	75.6	75.1	71.6
		1st derivative	69.4	79.2	80.0	75.9
		2nd derivative	66.8	77.5	80.1	73.9
		L_2-norm	67.5	77.9	78.5	74.5
		SNV-norm	70.5	76.8	79.5	75.8
	FRUP	Raw spectral	70.3	78.1	75.5	74.5
		1st derivative	72.8	75.1	75.5	75.8
		2nd derivative	69.8	70.9	73.6	71.5
		L_2-norm	73.5	78.6	77.0	76.8
		SNV-norm	73.2	85.2	81.2	80.0
	FRP	Raw spectral	68.4	80.2	78.6	75.5
		1st derivative	68.3	81.5	79.8	76.9
		2nd derivative	70.2	80.5	80.9	77.8
		L_2-norm	66.3	78.8	79.8	75.4
		SNV-norm	75.5	85.5	82.6	81.3
	THUP	Raw spectral	73.1	81.6	77.5	77.2
		1st derivative	72.0	82.1	76.2	77.0
		2nd derivative	74.3	81.6	77.8	78.2
		L_2-norm	76.3	82.5	77.1	78.9
		SNV-norm	80.2	84.6	85.6	83.1

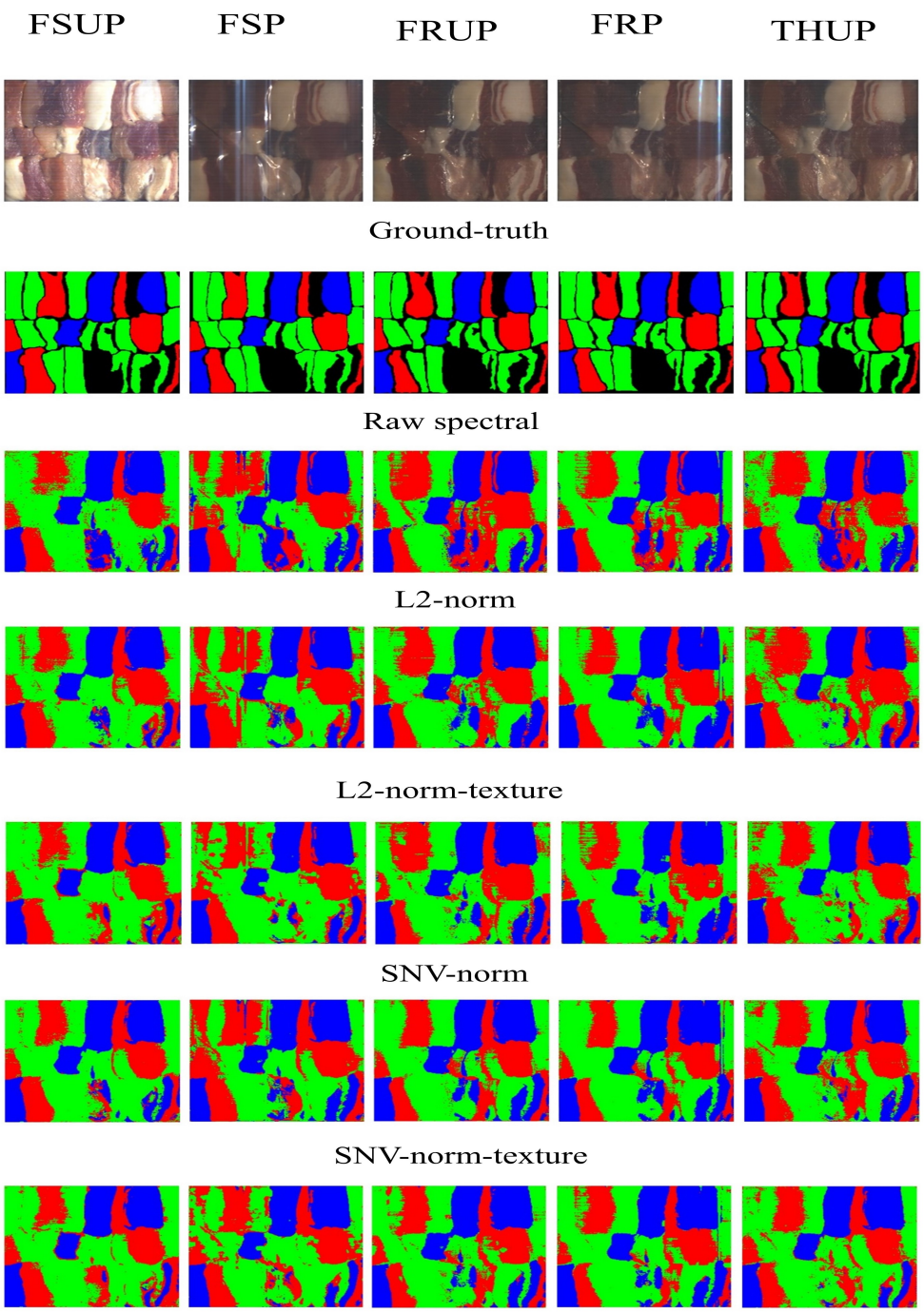


Figure A.1: Qualitative results of the SVM model by using different spectral and textural features. Red, Green, and Blue represent classes LAMB, OTHER (beef or pork), and FAT, respectively.

A.2 Supplementary material for chapter 5

This section provides a set of classification maps, resulting by applying the 3D–CNN model proposed in Chapter 5, for showing the robustness of the model in terms of classifying sequences of snapshot HSI images of meat samples; these sequences simulate the situation of video HSI imaging.

Figure A.2 shows loin chops of lamb and pork samples. These samples were scanned by the NIR snapshot HSI system, set of HSI sequences were collected to scan each sample. Also, Fig. A.2 shows the corresponding classification maps, from the 3D–CNN model, of these HSI sequences. The figure shows that the model is efficient in classifying the whole portions of samples and invariant to local changes of light sources. In Fig. A.3, the same meat samples (i.e., lamb and pork) were scanned by the VIS snapshot HSI system. The results in Fig. A.3 also shows the robustness of the 3D–CNN model for classifying the snapshot HSI sequences and showing the efficiency of the model in understanding both NIR and VIS hyperspectral image data.

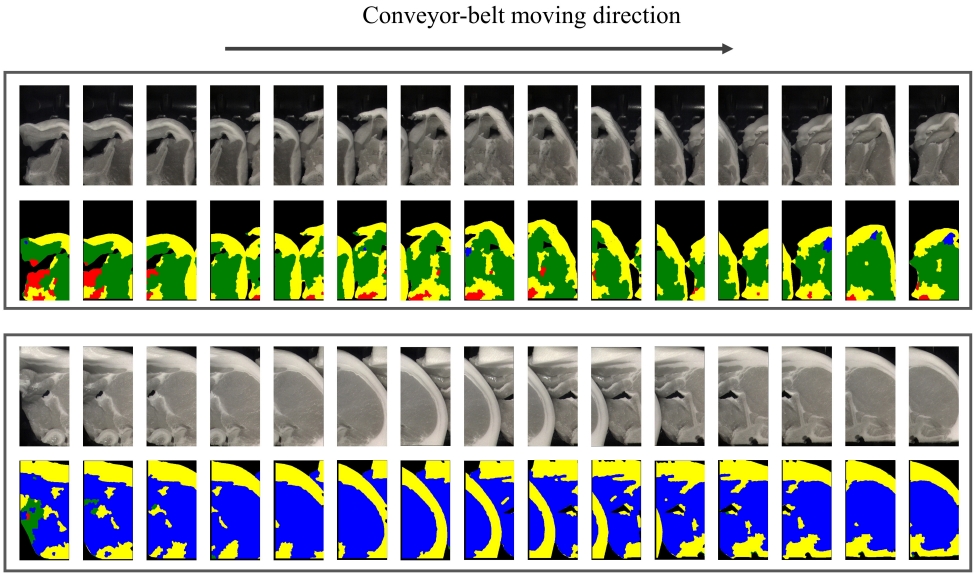


Figure A.2: Qualitative results of the proposed 3D–CNN model on NIR HSI images of meat samples. *Top*: Sequences of snapshot HSI of lamb (Loin chops) and their classification maps. *Bottom*: Sequences of snapshot HSI of pork (Loin chops) and their classification maps. Green, Red, Blue, and Yellow represent classes LAMB, BEEF, PORK, and FAT respectively.

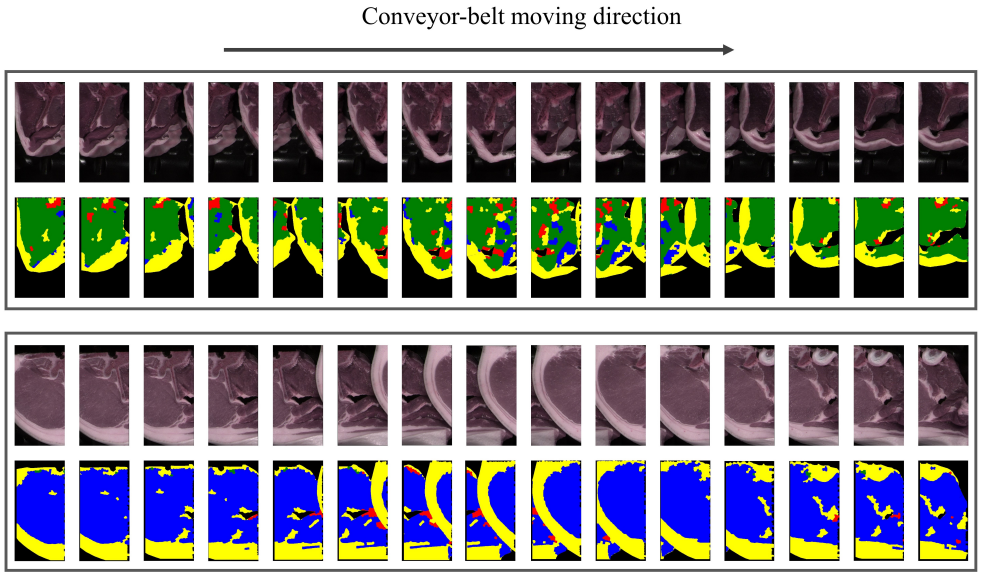


Figure A.3: Qualitative results of the proposed 3D–CNN model on VIS HSI images of meat samples. *Top*: Sequences of snapshot HSI of lamb (Loin chops) and their classification maps. *Bottom*: Sequences of snapshot HSI of pork (Loin chops) and their classification maps. Green, Red, Blue, and Yellow represent classes LAMB, BEEF, PORK, and FAT respectively.

A.3 Supplementary material for chapter 6

In general, deep learning models provide a property of accessing the learnt features of each layer of a deep learning model. These maps are commonly used for visualizing the output of each layer of a model. In this section, we visualize the resulting feature maps of the proposed FOD model and especially the proposed RPN model. These feature maps help in understanding the model and show that the model is efficient in understanding the HSI image contents (i.e., background, meat, and foreign objects).

Figure A.4, top, shows a selected HSI image form the dataset for visualization purposes. The image was contaminated by several types of foreign objects. After feeding the image into the proposed RPN model, selected features maps from the last layer of the first three CNN blocks were extracted and then plotted as shown in Fig. A.4, bottom.

The resulting feature maps, as shown in Fig. A.4, bottom, visually show that the FOD model learned a high level of spectral and spatial feature abstraction, where the foreign materials are highlighted in an intelligent way. Also, the model shows efficiency in discrimination between meat and the background based on their textural and spectral features.

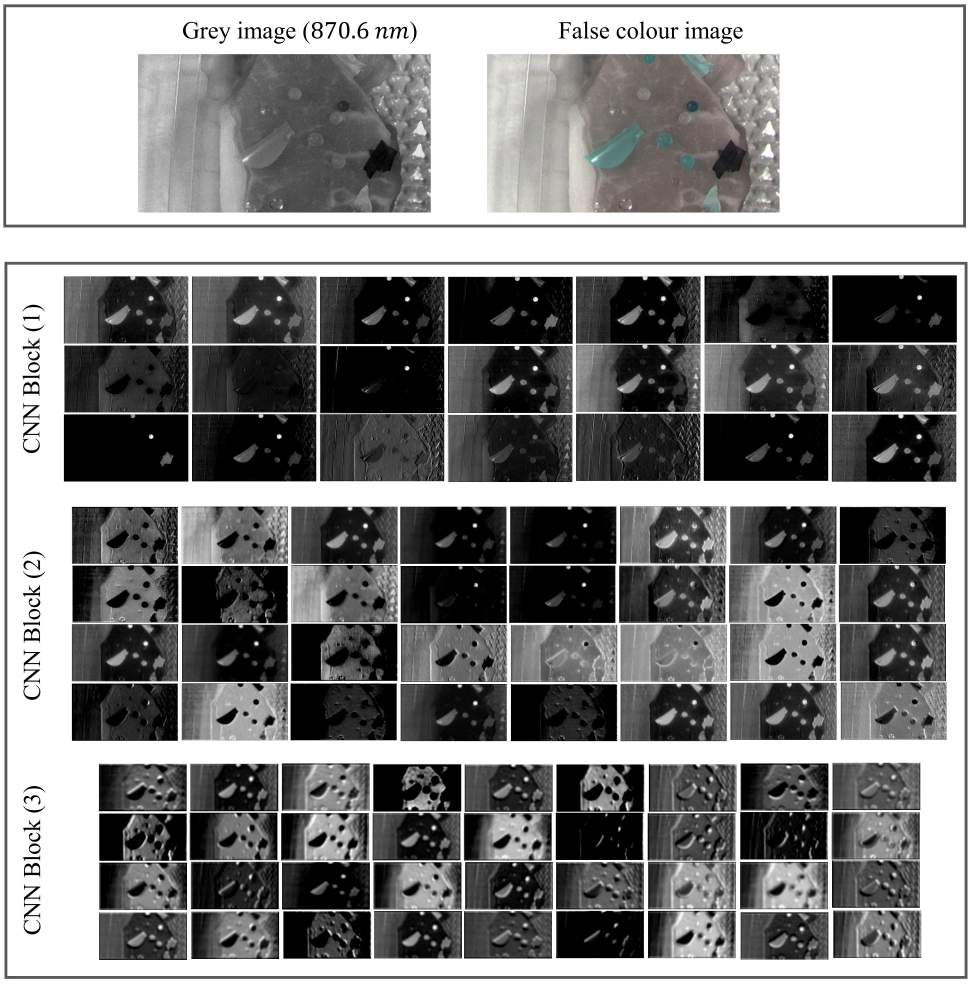


Figure A.4: Visualization of the resulting feature maps of the proposed FOD model for meat products. *Top*: Visualization of an input HSI image as grey and false colour images. *Bottom*: The resulting feature maps of the first three CNN blocks (selected maps from the last CNN layer of each block) for the input image.

Bibliography

[1] Mancini, R., Hunt, M.: Current research in meat color. *Meat Sci.* (2005) **71(1)**: 100–121

[2] Chen, K., Qin, C.: Segmentation of beef marbling based on vision threshold. *Comput. Electron. Agric.* (2008) **62**: 223–230.

[3] Rosenvold, K., Andersen, H.: Factors of significance for pork quality – a review. *Meat Sci.* (2003) **64**: 219–237.

[4] Otto, G., Roehe, R., Looft, H., Thoelking, L., Kalm, E.: Comparison of different methods for determination of drip loss and their relationships to meat quality and carcass characteristics in pigs. *Meat Sci.* (2004) **68**: 401–409.

[5] Andrés, S., Silva, A., Soares-Pereira, A., Martins, C., Bruno-Soares, A., Murray, I.: The use of visible and near infrared reflectance spectroscopy to predict beef M. longissimus thoracis et lumborum quality attributes. *Meat Sci.* (2008) **78**: 217–224.

[6] Warner, R., Kauffman, R., Greaser, M.: Muscle protein changes post mortem in relation to pork quality traits. *Meat Sci.* (1997) **45**: 339–352.

[7] Agullo, E., Centurion, M., Ramos, V., Bianchi, M.: Determination of total pigments in red meats. *J. Food Sci.* (1990) **55**: 250–251.

[8] Pathare, P., Opara, U., Al-Said, F.: Color measurement and analysis in fresh and processed foods: a review. *Food Bioprocess Technol.* (2013) **6**: 36–60.

[9] Bock, J., Connelly, R.: Innovative uses of near-infrared spectroscopy in food processing. *J. of Food Sci.* (2008) **73**: 91–98.

[10] Cen, H., He, Y.: Theory and application of near-infrared reflectance spectroscopy in determination of food quality. *Trends Food Sci. Technol.* (2007) **18**: 72–83.

[11] Reis, M., Van Beers, R., Al-Sarayreh, M., Shorten, P., QiYan, W., Saeys, W., Klette, R., Craigie, C.: Chemometrics and hyperspectral imaging applied to assessment of chemical, textural and structural characteristics of meat. *Meat Sci.* (2018) **144**: 100–109.

[12] Prevolnik, M., Andek-Potokar, M., škorjanc, D.: Predicting pork water-holding capacity with NIR spectroscopy in relation to different reference methods. *J. Food Eng.* (2010) **98**: 347–352.

[13] Rodbotten, R., Nilsen, B., Hildrum, K.: Prediction of beef quality attributes from early post mortem near infrared reflectance spectra. *Food Chemistry* (2000) **69**: 427–436.

[14] Geesink, G., Schreutelkamp, F., Frankhuizen, R., Vedder, H., Faber, N., Kranen, R., Gerritzen, M.: Prediction of pork quality attributes from near infrared reflectance spectra. *Meat Sci.* (2003) **65**: 661–668.

[15] Mengshi L., Cavinato, A., Mayes, D., Smiley, S., Huang, Y., Al-Holy, M., and Rasco, B.: Bruise detection in pacific pink salmon (Oncorhynchus gorbuscha) by visible and short-wavelength near-infrared (SWNIR) spectroscopy (600–1100 nm). *J. Agriculture Food Chemistry* (2003) **51**: 6404–6408.

[16] Viljoen, M., Hoffman, L., Brand, T.: Prediction of the chemical composition of freeze-dried ostrich meat with near infrared reflectance spectroscopy. *Meat Sci.* (2005) **69**: 255–261.

[17] Saranwong, S., Sornsrivichai, J., Kawano, S.: Prediction of ripe-stage eating quality of mango fruit from its harvest quality measured nondestructively by near infrared spectroscopy. *Postharvest Biology and Technol.* (2004) **31**: 137–145.

[18] Greensill, C., Newman, D.: An experimental comparison of simple NIR spectrometers for fruit grading applications. *Applied Eng. in Agriculture* (2001) **17**: 69–76.

[19] Elmasry, G., Kamruzzaman, M., Sun, D., Allen, P.: Principles and applications of hyperspectral imaging in quality evaluation of agro-food products: A review. *Critical Reviews Food Sci. Nutrition.* (2012) **52**: 999–1023.

[20] Lambe, N., Ross, D., Navajas, E., Hyslop, J., Prieto, N., Craigie, C., Bünger, L., Simm, G., Roehe, R.: The prediction of carcass composition and tissue distribution in beef cattle using ultrasound scanning at the start and/or end of the finishing period. *Livestock Sci.* (2010) **131**: 193–202.

[21] Kongsro, J., Røe, M., Kvaal, K., Aastveit, A., Egelandsdal, B.: Prediction of fat, muscle and value in Norwegian lamb carcasses using EUROP classification, carcass shape and length measurements, visible light reflectance and computer tomography (CT). *Meat Sci.* (2009) **81**: 102–107.

[22] Sahin, E., Yardimci, M., Cetingul, I., Bayram, I., Sengor, E.: The use of ultrasound to predict the carcass composition of live Akkaraman lambs. *Meat Sci.* (2008) **79**: 716–721.

[23] Orman, A., Çalıskan, G.Ü., Dikmen, S., Üstüner, H., Ogan, M., Çalıskan, Ç.: The assessment of carcass composition of Awassi male lambs by real-time ultrasound at two different live weights. *Meat Sci.* (2008) **80**: 1031–1036.

[24] Gresham, J., McPeake, S., Bernard, J., Henderson, H.: Commercial adaptation of ultrasonography to predict pork carcass composition from live animal and carcass measurements. *J. Animal Sci.* (1992) **70**: 631–639.

[25] Prieto, N., Navajas, E., Richardson, R., Ross, D., Hyslop, J., Simm, G., Roehe, R.: Predicting beef cuts composition, fatty acids and meat quality characteristics by spiral computed tomography. *Meat Sci.* **86**: 770–779.

[26] Du, C., Sun, D.: Comparison of three methods for classification of pizza topping using different colour space transformations. *J. Food Eng.* (2005) **68**: 277–287.

[27] Zheng, C., Sun, D., Zheng, L.: Recent developments and applications of image features for food quality evaluation and inspection–a review. *Trends in Food Sci. Technol.* (2006) **17**: 642–655.

[28] Wu, D., Sun, D.: Colour measurements by computer vision for food quality control: A review. *Trends in Food Sci. Technol.* (2013) **29**: 5–20.

[29] Knuth, D., Larrabee, T. L., Roberts, P., M.: *Mathematical Writing*, Mathematical Association of America, ISBN: 088385063X, 1989.

[30] Adão, T., Hruška, J., Pádua, L., Bessa, J., Peres, E., Morais, R., Sousa, J.: Hyperspectral imaging: A review on UAV-based sensors, data processing and applications for agriculture and forestry. *Remote Sensing* (2017) **9**: 1110.

[31] Lu, G., Fei, B.: Medical hyperspectral imaging: A review. *J. Biomedical Optics* (2014) **19**: 010901.

[32] ElMasry, G., Nakauchi, S.: Image analysis operations applied to hyperspectral images for non-invasive sensing of food quality–A comprehensive review. *Biosystems Eng.* (2016) **142**: 53–82.

[33] PWC company. Food fraud vulnerability assessment. Official technical report, available on-line at `https://www.pwc.co.nz/pdfs/pwc-food-fraud-vulnerability-assessment.pdf`, accessed in 18/10/2018.

[34] U.S. Food and Drug Administration (USA). Food Defense: Background and Global Importance. Official technical presentation, available on-line at `http://ficci.in/events/21353/ISP/P01_Food%20Defense%20Background%20&%20Global%20Importance.pdf`, accessed in 18/10/2018.

[35] Chinadaily. Well-known hot-pot restaurant involved in fake lamb scandal. News article, available on-line at `http://usa.chinadaily.com.cn/business/2013-05/06/content_16479947.htm`, accessed in 18/10/2018.

[36] Chinafile. Rat Meat Masquerading as Lamb—Yet Another Food Safety Scandal: News article, available on-line at `http://www.chinafile.com/reporting-opinion/media/rat-meat-masquerading-lamb-yet-another-food-safety-scandal`, accessed in 18/10/2018.

[37] DailyMail. Rat meat sold as lamb in Shanghai and tourists may have eaten it, police say. News article, available on-line at `https://www.dailymail.co.uk/news/article-2318957/Rat-meat-sold-lamb-Shanghai-tourists-eaten-police-say.html`, accessed in 18/10/2018.

[38] Interest. Chinese authorities investigating fake NZ lamb products sold in major chain hot pot restaurants. News article, available on-line at www.interest.co.nz/rural-news/64330/chinese-authorities-investigating-fake-nz-lamb-products-sold-ma\jor-chain-hot-pot-re, accessed in 18/10/2018.

[39] Food recall statistics, www.foodstandards.govt.nz/industry/foodrecalls/recallstats/\Pages/default.aspx. Last accessed 25 March (2019)

[40] MPI New Zealand. A guide to HACCP systems in the Meat Industry. Guidance Document, available on-line at www.mpi.govt.nz/dmsdocument/22081/send, accessed in 18/10/2018.

[41] Liu, F., He, Y., Wang, L., Sun, G.: Detection of organic acids and pH of fruit vinegars using near-infrared spectroscopy and multivariate calibration. *Food and Bioprocess Technology* (2011) **4**: 1331-1340.

[42] Kandpal, L., Lee, S., Kim, M., Bae, H., Cho, B.: Short wave infrared (SWIR) hyperspectral imaging technique for examination of aflatoxin B1 (AFB1) on corn kernels. *Food Control* (2015) **51**: 171-176.

[43] Gowen, A., Tiwari, B., Cullen, P., McDonnell, K., O'Donnell, C.: Applications of thermal imaging in food quality and safety assessment. *Trends in Food Science & Technology* (2010) **21**: 190-200.

[44] Barker, M., Rayens, W.: Partial least squares for discrimination. *J. of Chemometrics: A Journal of the Chemometrics Society* (2003) **17**: 166-173.

[45] Sun, D.: Hyperspectral imaging for food quality analysis and control. Elsevier, 2010.

[46] Khan, A., Munir, M., Yu, W., Young, B.: A Review Towards Hyperspectral Imaging for Real-Time Quality Control of Food Products with an Illustrative Case Study of Milk Powder Production. *Food Bioprocess Technol* (2020) **13**: 739–752.

[47] Wu, D., Sun, D.: Advanced applications of hyperspectral imaging technology for food quality and safety analysis and assessment: A review—Part I: Fundamentals. *Innovative Food Science Emerging Technologies* (2013) **19**: 1–14.

[48] Cheng, J., Nicolai, B., Sun, D.: Hyperspectral imaging with multivariate analysis for technological parameters prediction and classification of muscle foods: A review. *Meat Sci.* (2017) **123**: 182–191.

[49] Gonzalez, P. Geelen, B., Blanch, C., Tack, K., Lambrechts, A.: A CMOS-compatible, monolithically integrated snapshot-mosaic multispectral imager. *NIR News* (2015) **26**: 6–11.

[50] Geelen, B., Tack, N., Lambrechts, A.: A compact snapshot multispectral imager with a monolithically integrated per-pixel filter mosaic. *In Proc. SPIE 8974, Advanced Fabrication Technologies Micro/Nano Optics Photonics* (2014), 89740L.

[51] Liu, Y., Pu, H., Sun, D.: Hyperspectral imaging technique for evaluating food quality and safety during various processes: A review of recent applications. *Trends Food Science Technology* (2017) **69**: 25–35.

[52] Dey, N.: Uneven illumination correction of digital images: A survey of the state-of-the-art. *Optik* (2019) **183**: 483-495.

[53] West, M., Grossmann, J., Galvan, C.: Commercial Snapshot Spectral Imaging: the art of the possible. *The MITRE Corporation* (2019).

[54] Grusche, S.: Basic slit spectroscope reveals three-dimensional scenes through diagonal slices of hyperspectral cubes. *Applied Optics* (2014) **53**: 4594-4603.

[55] Liu, D., Zeng, X., Sun, D.: Recent developments and applications of hyperspectral imaging for quality evaluation of agricultural products: a review. *Critical Reviews Food Science Nutrition* (2015) **55**: 1744–1757.

[56] Teke, M., Deveci, H., Haliloğlu, O., Gürbüz, S., Sakarya, U.: A short survey of hyperspectral remote sensing applications in agriculture. *In Recent Advances in Space Technologies*. (2013) pp:171–176.

[57] Petersson, H., Gustafsson, D., Bergstrom, D.: Hyperspectral image analysis using deep learning: A review. *In Proc. IEEE Image Processing Theory Tools Applications* (2016) pp:1–6.

[58] Gewali, U., Monteiro, S., Saber, E.: Machine learning based hyperspectral image analysis: a survey. *ArXiv:1802.08701* (2018).

[59] Zhu, X., Tuia, D., Mou, L., Xia, G., Zhang, L., Xu, F., Fraundorfer, F.: Deep learning in remote sensing: a comprehensive review and list of resources. *IEEE Geoscience and Remote Sensing Magazine*. (2017) **5**: 8–36.

[60] Kamruzzaman,M., ElMasry, G., Sun, D., Allen, P.: Prediction of some quality attributes of lamb meat using near-infrared hyperspectral imaging and multivariate analysis. *Analytica Chimica Acta*. (2012) **714**: 57–67.

[61] ElMasry, G., Sun, D., Allen, P.: Near-infrared hyperspectral imaging for predicting colour, pH and tenderness of fresh beef. *J. Food Engineering* (2012) **110**: 127–140.

[62] Pu, H., Sun, D., Ma, J., Liu, D., Cheng, J.: Using Wavelet Textural Features of Visible and Near Infrared Hyperspectral Image to Differentiate Between Fresh and Frozen–Thawed Pork. *Food and Bioprocess Technology* (2014) **7**: 3088–3099.

[63] ElMasry, G., Sun, D., Allen, P.: Non-destructive determination of waterholding capacity in fresh beef by using NIR hyperspectral imaging. *Food Research International* (2011) **44**: 2624–2633.

[64] Kamruzzaman, M., Makino, Y., Oshita, S.: Rapid and Non-destructive detection of chicken adulteration in minced beef using visible near-infrared hyperspectral imaging and machine learning. *J. Food Engineering*. (2016) **170**: 8–15.

[65] Kamruzzaman, M., Elmasry, G., Sun, D., Allen, P.: Non-destructive assessment of instrumental and sensory tenderness of lamb meat using NIR hyperspectral imaging. *Food Chemistry* (2013) **141**: 389–396.

[66] Wu, J., Peng, Y., Li, Y., Wang, W., Chen, J., Dhakal, S.: Prediction of beef quality attributes using VIS/NIR hyperspectral scattering imaging technique. *J. Food Engineering* (2012) **109**: 267–273.

[67] Tao, F., Peng, Y.: A method for non-destructive prediction of pork meat quality and safety attributes by hyperspectral imaging technique. *J. Food Engineering* (2014) **126**: 98–106.

[68] Xiong, Z., Sun, D., Dai, Q., Han, Z., Zeng, X., Wang, L.: Application of visible hyperspectral imaging for prediction of springiness of fresh chicken meat. *Food Analytical Methods*. (2014) **8**: 380–391.

[69] Savitzky, A., Golay, A.: Smoothing and differentiation of data by simplified least squares procedures. *Anal. Chem.* (1964) **36**: 1627–1639.

[70] Norris, K.: Extracting information from spectrophotometric curves. Predicting chemical composition from visible and near-infrared spectra. *In Proc. IUFoST Symp. Food Res. Data Anal.* (1982) pp: 95–113.

[71] Feng, Y., Sun, D.: Near-infrared hyperspectral imaging in tandem with partial least squares regression and genetic algorithm for non-destructive determination and visualization of Pseudomonas loads in chicken fillets. *Talanta* (2013) **109**: 74–83.

[72] Barnes, R., Dhanoa, M., Lister, S.: Standard normal variate transformation and detrending of nearinfrared diffuse reflectance spectra. *Appl. Spectrosc.* (1989) **43**: 772–777.

[73] Wold, S., Antti, H., Lindgren, F., Öhman, J.: Orthogonal signal correction of near-infrared spectra. *Chemometrics and Intelligent Laboratory Systems* (1998) **44**: 175–185.

[74] Barbin, D., Elmasry, G., Sun, D., Allen, P.: Near-infrared hyperspectral imaging for grading and classification of pork. *Meat Sci.* (2012) **90**: 259–268.

[75] Jolliffe, I.: Principal component analysis. Springer, 2011.

[76] Fisher, R.: The use of multiple measurements in taxonomic problems. *Annals of Eugenics* (1936) **7**: 179-188.

[77] Kamruzzaman, M., ElMasry, G., Sun, D., Allen, P.: Application of NIR hyperspectral imaging for discrimination of lamb muscles. *J. Food Engineering* (2011) **104**: 332–340.

[78] Khojastehnazhand, M., Khoshtaghaza, M., Mojaradi, B., Rezaei, M., Goodarzi, M., Saeys, W.: Comparison of Visible–Near infrared and short wave infrared hyperspectral imaging for the evaluation of rainbow trout freshness. *Food Research International* (2014) **56**: 25–34.

[79] Kamruzzaman, M., Barbin, D., ElMasry, G., Sun, D., Allen, P.: Potential of hyperspectral imaging and pattern recognition for categorization and authentication of red meat. *Innovative Food Science & Emerging Technologies* (2012) **16**: 316–325.

[80] Wei, X., Liu, F., Qiu, Z., Shao, Y., He, Y.: Ripeness classification of astringent persimmon using hyperspectral imaging technique. *Food and Bioprocess Technology* (2014) **7**: 1371–1380.

[81] Ropodi, A. , Pavlidis, D., Mohareb, F., Panagou, E., Nychas, G.: Multispectral image analysis approach to detect adulteration of beef and pork in raw meats. *Food Research International* (2015) **67**: 12–18.

[82] Sanz, J., Fernandes, A., Barrenechea, E., Silva, S., Santos, V., Goncalves, N., Pater-nain, D., Jurio, A., Melo-Pinto, P.: Lamb muscle discrimination using hyperspectral imaging comparison of various machine learning algorithms. *J. Food Engendering* (2016) **174**: 92–100.

[83] Pu, H., Sun, D., Ma, J., Cheng, J.: Classification of fresh and frozen-thawed pork muscles using visible and near infrared hyperspectral imaging and textural analysis. *Meat Sci.* (2014) **99**: 81–88.

[84] Barbin, D., Sun, D., Su, C.: NIR hyperspectral imaging as non-destructive evaluation tool for the recognition of fresh and frozen–thawed porcine longissimus dorsi muscles. *Innovative Food Science & Emerging Technologies* (2013) **18**: 226–236.

[85] Kamruzzaman, M., Makino, Y., Oshita, S: Hyperspectral imaging in tandem with multivariate analysis and image processing for non-invasive detection and visualization of pork adulteration in minced beef. *Analytical Methods* (2015) **7**: 7496–7502.

[86] Daszkiewicz, T., Kubiak, D., Panfil, A.: The effect of long-term frozen storage on the quality of meat (Longissimus thoracis et lumborum) from female roe deer (Capreolus capreolus L.). *Journal of Food Quality* 2018.

[87] Zhang, C., Jiang, H., Liu, F., He, Y.: Application of near-infrared hyperspectral imaging with variable selection methods to determine and visualize caffeine content of coffee beans. *Food and Bioprocess Technology* (2017) **10**: 213–221.

[88] Ambroise, C., McLachlan, G.: Selection bias in gene extraction on the basis of microarray gene-expression data. *In Proc. Natl. Acad. Sci.* (2002) **99**: 6562–6566.

[89] Granitto, P.M., Furlanello, C., Biasioli, F., Gasperi, F.: Recursive feature elimination with random forest for PTR-MS analysis of agroindustrial products. *Chemom. Intell. Lab. Sys.* (2006) **83**: 83–90.

[90] Qu, J., Cheng, J., Sun, D., Pu, H., Wang, Q., Ma, J.: Discrimination of shelled shrimp (Metapenaeus ensis) among fresh, frozen-thawed and cold-stored by hyperspectral imaging technique. *LWT-Food Science and Technology* (2015) **62**: 202–209.

[91] Dai, Q., Cheng, J., Sun, D., Pu, H., Zeng, X., Xiong, Z.: Potential of visible/near-infrared hyperspectral imaging for rapid detection of freshness in unfrozen and frozen prawns. *J. Food Engineering* (2015) **149**: 97–104.

[92] Cheng, J., Sun, D., Pu, H., Chen, X., Liu, Y., Zhang, H., Li, J.: Integration of classifiers analysis and hyperspectral imaging for rapid discrimination of fresh from cold-stored and frozen-thawed fish fillets. *J. Food Engineering* (2015) **161**: 33–39.

[93] Sun, J., Wu, X., Zhang, X., Li, Q.: Identification of moisture content in tobacco plant leaves using outlier sample eliminating algorithms and hyperspectral data. *Biochemical Biophysical Research Communications* (2016) **471**: 226–232.

[94] Lohumi, S., Lee, S., Lee, H., Kim, M., Lee, W., Cho, B.: Application of hyperspectral imaging for characterization of intramuscular fat distribution in beef. *Infrared Physics & Technology* (2016) **74**: 1–10.

[95] Klette R. Concise computer vision. Springer, 2014.

[96] Gonzalez, R., Woods, R., Eddins, S.: Digital image processing using MATLAB. Gatesmark Publishing Knoxville, 2009.

[97] Comaniciu, D., Meer, P.: Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Analysis Machine Intelligence* (2002) **5**: 603–619.

[98] He, X., Zemel, R. S., Ray, D.: Learning and incorporating top-down cues in image segmentation. *In Proc. European conference on computer vision* (2006) pp. 338-351.

[99] Mori, G.: Guiding model search using segmentation. *In Proc. IEEE Int. Conf. Computer Vision* (2005) pp. 1417-1423.

[100] Fulkerson, B., Vedaldi, A., Soatto, S.: Class segmentation and object localization with superpixel neighborhoods. *In Proc. IEEE Int. Conf. Computer Vision* (2009) pp. 670-677.

[101] Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., Susstrunk, S.: SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Trans. Pattern Analysis Machine Intelligence* (2013) **34**: 2274–2282.

[102] Vedaldi, A., Soatto, S.: Quick shift and kernel methods for mode seeking. *In Proc. European Conference on Computer Vision* (2008) pp. 705–718.

[103] Felzenszwalb, P., Huttenlocher, D.: Efficient graph-based image segmentation. *IJCV* (2004) **59**: 167–181

[104] Levinshtein, A., Stere, A., Kutulakos, K., Fleet, D., Dickinson, S., Siddiqi, K.: Turbopixels: Fast superpixels using geometric flows. *IEEE Trans. Pattern Analysis Machine Intelligence* (2009) **31**: 2290–2297.

[105] Ghamisi, P., Couceiro, M., Benediktsson, J.: Integration of segmentation techniques for classification of hyperspectral images. *IEEE Geosci. Remote Sens. Lett.* (2014) **11**: 342–346.

[106] Huang, X., Zhang, L.: An adaptive mean-shift analysis approach for object extraction and classification from urban hyperspectral imagery. *IEEE Trans. Geosci. Remote Sens.* (2008) **46**: 4173–4185.

[107] Fang, L., Li, S., Duan, W.: Classification of hyperspectral images by exploiting spectral-spatial information of superpixel via multiple kernels. *IEEE Trans. Geosci. Remote Sens.* (2015) **53**: 6663–6674.

[108] Haralick, R.: Statistical and structural approaches to texture. *Proceedings of the IEEE* (1979) **67**: 786–804.

[109] Naganathan, G., Cluff, K., Samal, A., Calkins, C., Jones, D., Lorenzen, C., Subbiah, J.: Hyperspectral imaging of ribeye muscle on hanging beef carcasses for tenderness assessment. *Computers Electronics Agriculture* (2015) **116**: 55–64.

[110] Yang, J., Zhang, D., Frangi, A. F., Yang, J. Y.: Two-dimensional PCA: a new approach to appearance-based face representation and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. (2004) **26**: 131-137.

[111] Guo, T., Huang, M., Zhu, Q., Guo, Y., Qin, J.: Hyperspectral image-based multi-feature integration for TVB-N measurement in pork. *J. Food Engineering* (2018) **218**: 61–68.

[112] Zhang, L., Du, B.: Deep learning for remote sensing data: A technical tutorial on the state of the art. *IEEE Geosci. Remote Sens. Magazine* (2016) **4**: 22–40.

[113] Pu, H., Sun, D., Ma, J., Liu, D., Cheng, J.: Using wavelet textural features of visible and near infrared hyperspectral image to differentiate between fresh and frozen–thawed pork. *Food Bioprocess Technology* (2014) **7**: 3088–3099.

[114] Ivorra, E., Girón, J., Sánchez, A., Verdú, S., Barat, J., Grau, R.: Detection of expired vacuum-packed smoked salmon based on PLS-DA method using hyperspectral images. *J. Food Engineering* (2013) **117**: 342–349.

[115] Cortes, C., Vapnik, V.: Support-vector networks. *Machine Learning* (1995) **20**: 273–297.

[116] Ravikanth, L., Singh, C., Jayas, D., White, N.: Classification of contaminants from wheat using near-infrared hyperspectral imaging. *Biosystems Engineering* (2015) **135**: 73–86.

[117] Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press, 2016.

[118] Krizhevsky, A., Sutskever, I., Hinton, G.: ImageNet classification with deep convolutional neural networks. *In Proc. Advances Neural Information Processing Systems* (2012) pp. 1097–1105.

[119] Ciregan, D., Meier, U., Schmidhuber, J.: Multi-column deep neural networks for image classification. *In Proc. Computer Vision Pattern Recognition* (2012) pp. 3642–3649.

[120] Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. *In Proc. Computer Vision Pattern Recognition* (2014) pp. 580–587.

[121] Taigman, T., Yang, M., Ranzato, M., Wolf, L.: DeepFace: closing the gap to human-level performance in face verification. *In Proc. Computer Vision Pattern Recognition* (2014) pp. 1701–1708.

[122] Shuiwang, J., Wei, X., Ming, Y., Kai, Y.: 3D convolutional neural networks for human action recognition. *IEEE Trans. Pattern Analysis Machine Intelligence* (2013) **35**: 221–231.

[123] Sainath, T.N., Mohamed, A.R., Kingsbury, B., Ramachandran, B.: Deep convolutional neural networks for LVCSR. *In Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing* (2013) pp. 8614–8618.

[124] Abdel-Hamid, O., Mohamed, A.R., Jiang, H., Penn, G.: Applying convolutional neural networks concepts to hybrid NN-HMM model for speech recognition. *In Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing* (2012) pp. 4277–4280.

[125] Ji, S., Zhang, C., Xu, A., Shi, Y., Duan, Y.: 3D Convolutional Neural Networks for Crop Classification with Multi-Temporal Remote Sensing Images. *J. Remote Sens.* (2018) 10.

[126] Halicek, M., Lu, G., Little, J., Wang, X., Patel, M., Griffith, C., El-Deiry, M., Chen, A., Fei, B.: Deep convolutional neural networks for classifying head and neck cancer using hyperspectral imaging. *J. Biomedical Optics* (2017) **22**: 060503.

[127] Chen, Y., Zhao, X., Jia, X.: Spectral–spatial classification of hyperspectral data based on deep belief network. *IEEE J. Sel. Top. Applied Earth Observations Remote Sens.* (2015) **8**: 2381–2392.

[128] Chen, Y., Lin, Z., Zhao, X., Wang, G., Gu, Y.: Deep learning-based classification of hyperspectral data. *IEEE J. Sel. Top. Applied Earth Observations Remote Sens.* (2014) **7**: 2094–2107.

[129] Hu, W., Huang, Y., Wei, L., Zhang, F., Li, H.: Deep convolutional neural networks for hyperspectral image classification. *J. Sens*. (2015) 12.

[130] Chen, Y., Jiang, H., Li, C., Jia, X., Ghamisi, P.: Deep feature extraction and classification of hyperspectral images based on convolutional neural networks. *IEEE Trans. Geosci. Remote Sens*. (2016) **54**: 6232–6251.

[131] Li, Y., Zhang, H., Shen, Q.: Spectral–spatial classification of hyperspectral imagery with 3D convolutional neural network. *J. Remote Sens*. (2017) 9.

[132] Li, W., Wu, G., Zhang, F., Du, Q.: Hyperspectral image classification using deep pixel-pair features. *IEEE Trans. Geosci. Remote Sens.* (2017) **55**: 844–853.

[133] Li, Y., Xie, W., Li, H.: Hyperspectral image reconstruction by deep convolutional neural network for classification. *Pattern Recognition* (2017) **63**: 371–383.

[134] Hamida, A., Benoit, A., Lambert, P., Amar, C.: 3-D deep learning approach for remote sensing image classification. *IEEE Trans. Geosci. Remote Sens.* (2018) **56**: 4420–4434.

[135] Luo, Y., Zou, J., Yao, C., Zhao, X., Li, T., Bai, G.: HSI-CNN: A novel convolution neural network for hyperspectral image. *In Proc. Int. Conf. Audio Language Image Processing* (2018) pp. 464- -469.

[136] Roy, S., Krishna, G., Dubey, S., Chaudhuri, B.: HybridSN: Exploring 3D-2D CNN feature hierarchy for hyperspectral image classification. *ArXiv:1902.06701* (2019).

[137] He, M., Li, B., Chen, H.: Multi-scale 3d deep convolutional neural network for hyperspectral image classification. *In Proc. IEEE Int. Conf. Image Processing* (2017) pp. 3904-3908.

[138] Mou, L., Ghamisi, P., Zhu, X.: Deep recurrent neural networks for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* (2017) **55**: 3639–3655.

[139] Liu, Q., Zhou, F., Hang, R., Yuan, X.: Bidirectional-convolutional LSTM based spectral-spatial feature learning for hyperspectral image classification. *Remote Sens.* (2017) 9.

[140] Karrer, A., Stuart, A., Craigie, C., Taukiri, K., Reis, M.: Detection of adulteration in meat product using of hyperspectral imaging. *In Proc. Chemometrics Analytical Chemistry* (2016) pp. 177.

[141] Hu, Y., Monteiro, S., Saber, E.: Super pixel based classification using conditional random fields for hyperspectral images. *In Proc. IEEE Int. Conf. Image Processing* (2016) pp. 2202–2205.

[142] Kennard, R., Stone, L.: Computer aided design of experiments. *Technometrics* (1969) **11**: 137–148.

[143] Svetnik, V., Liaw, A., Tong, C., Wang, T.: Application of Breiman's random forest to modeling structure activity relationships of pharmaceutical molecules. *In Proc. Int. Workshop Multiple Classifer Systems* (2004) pp. 334–343.

[144] Maaten, L., Hinton, G.: Visualizing data using t-SNE. *J. Machine Learning Research* (2008) **9**: 2579–2605.

[145] Namin, S., Petersson, L: Classification of materials in natural scenes using multi-spectral images. *In Proc. IEEE Int. Conf. Intelligent Robots Systems* (2012) pp. 1393–1398.

[146] Bradley, D., Unnikrishnan, R., Bagnell, J.: Vegetation detection for driving in complex environments. *In Proc. IEEE Int. Conf. Robotics Automation* (2007) pp. 503–508.

[147] Winkens, C., Sattler, F., Paulus, D.: Hyperspectral terrain classification for ground vehicles. *In Proc. Int. Joint Conf. Computer Vision, Imaging and Computer Graphics Theory and Applications* (2017) pp. 417–424.

[148] Winkens, C., Kobelt, V., Paulus, D.: Robust features for snapshot hyperspectral terrain-classification. *In Proc. Int. Conf. Computer Analysis Images Patterns* (2017) pp. 16–27.

[149] Ishida, T., Kurihara, J., Viray, F., Namuco, S., Paringit, E., Perez, G., Takahashi, Y., Marciano, J.: A novel approach for vegetation classification using UAV-based hyperspectral imaging. *Computers Electronics Agriculture* (2018) **144**: 80–85.

[150] Mahesh, S., Jayas, D., Paliwal, J., White, N.: Identification of wheat classes at different moisture levels using near-infrared hyperspectral images of bulk samples. *Sensing Instrumentation Food Quality Safety* (2011) **5**: 1–9.

[151] Serranti, S., Cesare, D., Marini, F., Bonifazi, G.: Classification of oat and groat kernels using NIR hyperspectral imaging. *Talanta* (2013) **103**: 276–284.

[152] Wang, O., Gunawardane, P., Scher, S., Davis, J.: Material classification using BRDF slices. *In Proc. IEEE Conf. Computer Vision Pattern Recognition* (2009) pp. 2805–2811.

[153] Tatzer, P., Wolf, M., Panner, T.: Industrial application for inline material sorting using hyperspectral imaging in the NIR range. *Real-Time Imaging* (2005) **11**: 99–107.

[154] Fotiadou, K., Tsagkatakis, G., Tsakalides, P.: Deep convolutional neural networks for the classification of snapshot mosaic hyperspectral imagery. *Electronic Imaging* (2017) **17**: 185–190.

[155] Swatland, H. J.: Meat Cuts and Muscle Foods: An International Glossary. Nottingham University Press, 2004.

[156] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: A simple way to prevent neural networks from overfitting. *J. Machine Learning Research* (2014) **15**: 1929–1958.

[157] Kingma, D., Ba, J.: Adam: A method for stochastic optimization. *ArXiv:1412.6980* (2014).

[158] Edwards, M.: Detecting Foreign Bodies in Food. Elsevier (2004).

[159] Graves, M., Smith, A., Batchelor, B. Approaches to foreign body detection in foods. *Trends Food Science Technology* (1998) **9**: 21–27.

[160] Nielsen, M., Lauridsen, T., Christensen, L., Feidenhans, R.: X-ray dark-field imaging for detection of foreign bodies in food. *Food Control* (2013) **30**: 531–535.

[161] Chen, X., Jing, H., Tao, Y., Cheng, X.: High-resolution real-time x-ray and 3D imaging for physical contamination detection in deboned poultry meat. *In Proc. SPIE Monitoring Food Safety Agriculture Plant Health* (2004) pp. 108-118.

[162] Einarsdóttir, H., Emerson, M. J., Clemmensen, L. H., Scherer, K., Willer, K., Bech, M., Larsen, R., Ersbøll, B., Pfeiffer, F.: Novelty detection of foreign objects in food using multi-modal X-ray imaging. *Food Control* (2016) **67**: 39–47.

[163] Feng, C.-H., Makino, Y., Oshita, S., Martín, J.-F.: Hyperspectral imaging and multispectral imaging as the novel techniques for detecting defects in raw and processed meat products: Current state-of-the-art research advances. *Food Control* (2018) **84**: 165–176.

[164] Vejarano, R., Siche, R., Tesfaye, W.: Evaluation of biological contaminants in foods by hyperspectral imaging: A review. *Int. J. Food Properties* (2017) **20**: 1264–1297.

[165] Cao, X., Wang, P., Meng, C., Bai, X., Gong, G., Liu, M., Qi, J.: Region based CNN for foreign object debris detection on airfield pavement. *Sensors* (2018) 18, 737.

[166] Han, Z., Fang, Y., Xu, H., Zheng, Y.: A novel FOD classification system based on visual features. *In: Proc. Int. Conf. Image Graphics* (2015) pp. 288–296.

[167] Xu, H., Han, Z., Feng, S., Zhou, H., Fang, Y.: Foreign object debris material recognition based on convolutional neural networks. *Eurasip J. Image Video Processing* (2018) 1, 21.

[168] Chalapathy, R., Menon, A. K., Chawla, S.: Anomaly detection using one-class neural networks. *ArXiv:1802.06360* (2018).

[169] Sommer, C., Hoefler, R., Samwer, M., Gerlich, D. W.: A deep learning and novelty detection framework for rapid phenotyping in high-content screening. *Molecular Biology Cell* (2017) **28**: 3428–3436 (2017)

[170] Lowe, D.-G.: Distinctive image features from scale-invariant keypoints. *Int. J. Computer Vision* (2004) **60**: 91–110.

[171] Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. *In Proc. IEEE Conf. Computer Vision Pattern Recognition* (2005) pp. 886–893.

[172] Ojala, T., Pietikäinen, M., Mäenpää, T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans. Pattern Analysis Machine Intelligence* (2002) **7**: 971–987.

[173] Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. *ArXiv:1409.1556* (2014)

[174] Girshick, R.: Fast R-CNN. *In Proc. IEEE Int. Conf. Computer Vision* (2015) pp. 1440–1448.

[175] Ren, S., He, K., Girshick, R.-B., Sun, J.: Faster R-CNN: Towards real-time object detection with region proposal networks. *In Proc. Advances Neural Information Processing Systems* (2015) pp. 91–99.

[176] He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN. *In Proc. IEEE Int. Conf. Computer Vision* (2017) pp. 2961–2969.

[177] Jiang, H., Learned-Miller, E.: Face detection with the faster R-CNN. *In Proc. IEEE Int. Conf. Automatic Face Gesture Recognition* (2017) pp. 650–657.

[178] Uijlings, J.-R., Van De Sande, K.-E., Gevers, T., Smeulders, A.-W.: Selective search for object recognition. *Int. J. Computer Vision* (2013) **104**: 154–171.

[179] Redmon, J., Divvala, S.-K., Girshick, R.-B., Farhadi, A.: You only look once: Unified, real-time object detection. *In Proc. IEEE Conf. Computer Vision Pattern Recognition* (2016) pp. 779–788.

[180] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C., Berg, A.C.: SSD: Single shot multibox detector. *In Proc. European Conf. Computer Vision* (2016) pp. 21–37.