1859

ScuDo
Scuola di Dottorato ~ Doctoral School
WHAT YOU ARE, TAKES YOU FAR

UNIVERSITÀ
DEGLI STUDI
DI TORINO

Doctoral Dissertation
Doctoral Program in Urban and Regional Development (XXIII cycle)

Service robotics and machine learning for close-range remote sensing along real applications

Maria Angela Musci

* * * * * *

Supervisors
Prof. Andrea Maria Lingua, Supervisor
Eng. Paolo Dabove, Co-Supervisor
Eng. Irene Aicardi, Co-Supervisor

Doctoral Examination Committee:
Prof. Domenico Sguerso, Referee, Università di Genova
Prof. Domenico Visintini, Referee, Università degli studi di Udine
Prof. ssa Eufemia Tarantino, Politecnico di Bari
Dr. Eng. Rodrigo Gonzalez, GridTics. Universidad Tecnológica Nacional
Prof. Fabio Giulio Tonolo, Politecnico di Torino

Politecnico di Torino
April 2021

Funding: This research was funded by the PoliTO Interdepartmental Centre for Service Robotics (PIC4SeR).

PIC4SeR
PoliTO Interdipartimental Centre for Service Robotics

I hereby declare that the contents and organisation of this dissertation constitute my own original work and does not compromise in any way the rights of third parties, including those relating to the security of personal data.

…………………………………..…

Maria Angela Musci
Turin, 2020

Abstract

In the management of urban and regional development, continuous environmental monitoring plays a relevant role because of its impacts on productivity, resilience, and sustainability. Among the tools for data-gathering and data-analysis, service robotics and machine learning are becoming widely adopted, allowing a deep understanding of the environment and its processes. Their use enables new methods for managing urban and regional development. Nevertheless, the state of the art of platforms, sensors, and automatic information extraction techniques highlights that technologies and methodologies for monitoring might be further enhanced. Thus, improvements are required concerning the acquisition and the rapid analysis of high spatial, temporal, and spectral resolution data.

In this scenario, the interdepartmental centre of Politecnico di Torino for Service Robotics (PIC4SeR[1]), within which the present research work develops, aims at integrating service robotics and artificial intelligence in several applications as precision agriculture, cultural heritage, and smart cities & search & rescue. In these application fields, the Centre's idea stresses the development of a multi-agent and multi-sensor uncrewed platforms collaborating among themselves and enabling various perspectives for monitoring and interacting with the environment. This thesis, funded by the Centre, focuses on the geomatics aspects and consists of platform definition, sensors calibration, and definition of quasi real-time machine learning algorithms for information extraction. Two complex case studies, aircraft inspection for de-icing operations and precision agriculture, were investigated to test technologies and methods.

The purpose of the research is to define quasi-real-time automated information extraction techniques applied to multi-scale, multitemporal and multi-sensor data

[1] https://pic4ser.polito.it/

for inspection and environmental monitoring. Real-time automated techniques can make data analysis time-effective and more efficient for different applications.

The research spans the whole process of remote sensing, from platform definition to information extraction. In detail, the present work focuses on Uncrewed Aerial Vehicle (UAV) platform configuration to integrate innovative sensors such as a hyperspectral camera. At the same time, the research tests methodologies for extracting information using machine learning techniques from data with different spectral and spatial resolution such as Red, Green, Blue (RGB), multispectral and hyperspectral images.

Among the obtained promising results, it is possible to underline the following main findings. A UAV platform was configurated and fully integrated with the hyperspectral sensor and the correlated hardware components, focusing in particular on the ice detection inspection. The hyperspectral sensor was characterized, and the data were corrected geometrically and radiometrically. A suitable dimensionality reduction procedure for hyperspectral data was performed, showing the advantages of data management. Reduced hyperspectral and multispectral data were compared to verify the goodness of their adoption. Moreover, the band reduction procedure provided helpful information for selecting a multispectral sensor with equivalent bands. The advantages that this outcome provides is great because the multispectral camera is cheaper and simpler than the hyperspectral camera. This prospects further developments for near- real-time application in which spectral information could be essential.

In the end, a near-real time detection task was accomplished using semantic segmentation and object detection techniques with different data types (RGB, multispectral and hyperspectral). The outcomes show a promising model generalization with high accuracy values (80%-95%) in both applications, i.e. precision agriculture and aircraft inspection for de-icing operations. The results of this work can be extended to several UAV application fields, offering improved methods in near real-time object detection.

Acknowledgement

Throughout the development of this research, I have received a great deal of support and assistance. I am deeply grateful to Politecnico di Torino Interdepartmental Centre for Service Robotics (PIC4SeR) for financing my thesis. I would like to express special thanks to the PhD students and the researchers of PIC4SeR for the collaboration and the support. I would like to acknowledge the "Azienda Agricola Ciabot", and the "Dipartimento di Scienze Agrarie, Forestali e Alimentari" (DISAFA) of Università di Torino for giving me the chance to use their vineyards as study cases.

I would like to extend my sincere thanks to the Department of Environment, Land and Infrastructure Engineering (DIATI) of Politecnico di Torino where I worked since my Master thesis, and I developed this thesis. During these years, I had the privilege to know numerous people, colleagues of the Geomatic Research Group, who have contributed to my work and make me feel part of a group. A special thanks to my friend and office-mate Vincenzo di Pietra, for fun, support, and creative brainstorming, Nives Grasso for her advises and help and Francesca Noardo, for her suggestions during these years, and her support during my foreign experience. Thanks to my colleagues, Elena Belcore and Francesca Matrone, for sharing knowledge and this PhD experience. I wish to thank Paolo Maschio, Stefano Angeli for helping me with the data acquisition, and Luigi Mazzara to support the activities related to the hyperspectral data processing. I would like to offer my special thanks to Prof. Marco Piras, for encouraging me for all these years, I owe my passion for the research to him.

I would like to express my sincere gratitude to the Geomatic Research Group of the Department of Architecture and Design (DAD) colleagues, with I had the pleasure to collaborate for the activities related to the hyperspectral camera. Special

thanks go to my colleague, Alessio Calantropio, who helped me on many occasions, as DJI student ambassador and a friend.

I have been privileged to visit the Department of Earth Observation Science (EOS) at the University of Twente, Faculty of Geo-Information Science and Earth Observation (ITC) in 2019. I want to express my appreciation to Prof. Dr Ir. Alfred Stein, who offered me the opportunity to collaborate with the ITC, and Prof. Claudio Persello, who shared his knowledge about machine and deep learning and helped me to develop part of this research. Special thanks go, also, to my friends and colleagues at ITC, Anurag Kulshrestha, John Ray Bergado, Abhisek Maiti, and Wufan Zhao, for the helpful collaboration, the share of their knowledge, whose together with Surakshya Dhakal, Zehua Zhang and Rosa Maria Aguilar always made me feel welcome during my time in Enschede.

I am also very grateful to the "Centro Nazionale di Ricerca, Istituto per il rilevamento Elettromagnetico dell'Ambiente" (CNR-IREA) of Milan, and especially to Dr Daniela Stoppiana, Dr Paolo Villa for supporting in the activities related to the radiometric inter-calibration of the hyperspectral camera. Thanks also for the suggestions and sharing of technical knowledge.

I would like to acknowledge the working groups of the projects in which I was involved during these years, Advanced Forest Fire Fighting (AF3), Risk evaluation Dashboard (RED), RocktheAlps, Spectral Evidence of the Ice (SEI) and Vitifuture and Bee Monitoring in Natural Systems (BEEMS). I owe to these collaborations my work experience with interdisciplinary groups.

My appreciation also goes to Dr Gianpaolo Priore to share literature and help me with his knowledge related to vineyards. Thanks to my friends and colleagues Maddalena Marchelli, and Valerio De Biagi to advise and encourage me during this challenging period.

Maria Angela Musci
Turin, 2020

I want to dedicate this thesis to my family, Gina, Salvatore, Maria Nunzia, Andrea, Giulia, Daniele, Anna, Corrado, Maria, Carmelo e Gaspare, and to my loving husband, Antonio.

Thanks to them, I found the strength to finish this journey and believe in myself.

Laughing made us
invincible, not like those
who always win, but like
those that never give
up

Frida Kahlo

Contents

List of figures

List of tables

List of Abbreviations

3-D	Three-Dimensional
4G	Fourth Generation
AI	Artificial intelligence
ANN	Artificial Neural Network
AP	Average precision
AR	Augmented Reality
ARTM	Atmospheric Radiative Transfer Model
BBA	Bundle Block Adjustment
BRDF	Bidirectional Reflectance Distribution Function
BRISK	Binary Robust Invariant Scalable Keypoints
BHT	Binary Hierarchical Tree
BHT-BB	BHT-Balanced Branches Strategy
BHT-OAA	BHT-One Against All Strategy
C	Changeable
CART	Classification And Regression Tree
CBS	Constrained Band Selection
CCD	Charge-Coupled Device
CMOS	Complementary Metal Oxide Semiconductor
CNN	Convolutional Neural Network
CNR IREA	Centro Nazionale di Ricerca, Istituto per il rilevamento Elettromagnetico dell'Ambiente
CV	Computer Vision
DN	Digital Number
DN_N	Digital Number Normalized
DSM	Digital Surface Model
DSNU	Dark Signal Non-Uniformity
DT	Decision Tree

EG	Ethylene Glycol
ELM	Empirical Line Method
ENVI	Exelis Visual Information Solutions
EU	European Union
FAA	Federal Aviation Administration
FASTER R-CNN	Faster Region Based Convolutional Neural Network
FCNN	Fully Convolutional Neural Network
FD	Flavescence Dorée
FE	Feature Extraction
FIR	Far Infrared
FOV	Field of view
FPI	Fabry-Perrot Interferometer
FS	Feature Selection
FWHM	Full Width at Half Maximum
GA	Genetic Algorithm
GCP	Ground Control Point
GCS	Ground Control Station
GeoAI	Geospatial Artificial intelligence
GEOBIA	Geographic Object-Based Approaches
GIS	Geographical Information System
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GTP	Ground-Truth Point
GSD	Ground Sample Distance
HDRF	Hemispherical-Directional Reflectance Factors
HFOV	Horizontal Field Of View
HR	High Resolution
HSF	Harris-Stephens Features
HSI	Hyperspectral Imaging Sensor
ICA	Independent Component Analysis
ICT	Information and Communication Technology
IDBS	Information Divergence Band Selection
IE	Information Extraction
IFOV	Instantaneous Field Of View
IMU	Inertial Measurement Unit
IN	In-built

IOP	Internal Orientation Parameter
IOU	Intersection Over Union
IP	Internet Protocol
K-NN	K-Nearest Neighbors
LARS	Low Altitude Remote Sensing
LCMV	Linearly Constraint Minimum Variance
LDA	Linear Discriminant Analysis
LF	Large Format
LiDAR	Light Detection and Ranging
LiPO	Lithium Polymer battery
LMS	Laboratory for Manufacturing Systems and Automation
LoD	Levels of Details
LPBS	Linear-Prediction-based Band Selection
L_s	At-Sensor Radiance
LTE	Long Term Evolution
LWIR	Long Wavelength Infrared
MCC	MAPIR Camera Control
MDI	Minimum Dependent Information
MESMA	Multiple Endmember Spectral Mixture Analysis
MEV	Maximum Ellipsoid Volume
MF	Medium Format
ML	Machine Learning
MI	Maximum Information
MIR	Medium Wavelength Infrared
MR	Manifold Ranking
MSER	Maximally Stable Extremal Regions
MSI	Multispectral Imaging Sensor
MSN	Mobile Sensor Network
MVPCA	Maximum-Variance Principal Component Analysis
NASA	National Aeronautics and Space Administration
NIR	Near-Infrared
NMS	Non-Maximum Suppression
NRTK	Network Real-Time Kinematic
OA	Overall Accuracy
OAA	One-Against-All
OAO	One-Against-One

OB	Object-Based
OBIA	Object-based Approaches
OIF	Optimal Index Factor
OOB	Out-of- Bag
PA	Precision Agriculture
PAN	Panchromatic
PC	Principal Component
PCA	Principal Component Analysis
PG	Propylene Glycol
PIC4SeR	PoliTO Interdepartmental Centre for Service Robotics
PLSR	Partial least Squares Regression
PP	Principal Point
PWM	Pulse Width Modulation
R	Reflectance factor
R-CNN	Region Based Convolutional Neural Network
RADAR	Radio Detection and Ranging
RF	Random Forest
RGB	Red, Green, Blue
RGBN	Red, Green, Blue and Near-Infrared
RGN	Red, Green, and Near-Infrared
RL	Reinforcement Learning
ROI	Region of Interest
RP	Reference Panel
RPN	Region Proposal Network
RS	Remote Sensing
RTF	Ready-to-fly
RTK	Real-Time Kinematic
SBS	Sequential Backward Selection
SDG	Sustainable Development Goal
SDK	Software Development Kit
SF	Small-Format
SFFS	Sequential Floating Forward Selection
SfM	Structure from motion
SEI	Spectral Evidence of the Ice
SIFT	Scale Invariant Feature Transformation
SONAR	Sound Navigation and Ranging

SRF	Spectral Response Function
SSD	Single Shot MultiBox Detector
SSP-NET	Spatial Pyramid Pooling Network
SSN	Static Sensor Network
SURF	Speeded-Up Robust Features
SVM	Support Vector Machine
SWIR	Short Wavelength Infrared
UA	User’s Accuracy
UAS	Uncrewed Aerial System
UAV	Uncrewed Aerial Vehicle
UGS	Uncrewed Ground Systems
UGV	Uncrewd Ground Vehicles
URD	Urban and Regional Development
VGBS	Volume-Gradient-based Band Selection
VFOV	Vertical Field of View
VHF	Very High Frequency
VNIR	Visible to NIR
WLAN	Wide Local Area Network
YOLO	You Only Look Once

Chapter 1

Introduction

Managing monitoring and inspection at multiple scales (global, regional and local) is a complex issue in many real-world applications that imply maintenance and sustainability. This was a relevant task in the past, and it plays an important role also nowadays. Natural ecosystems [1], urban environment monitoring [2],[3],[4] precision agriculture [5], [6] and all the fields in which reducing the pollutants usage is essential are a few examples [7], [8] of how the inspection deeply enters in various activities at different scales.

Monitoring refers to the production of observations that enables the study, the inspection, and the understanding of the environment at different levels of details (LoDs). Monitoring and inspection of assets, and of the environment encompass repetitive and straightforward activities, i.e. survey, localization, object detection that should be done multiple times in different periods. However, they can be time-consuming, expensive and risky. In many of the above applications, that include, e.g., Precision Agriculture (PA) or farming, these tasks are accomplished with the involvement of experts in manual mode [9], [10] or by using static sensor networks (SSNs). Both of these approaches imply limited spatial and temporal coverage [11].

Robots and autonomous systems, also known as mobile sensor networks (MSNs) [12], can fulfill these tasks in a more efficient way improving safety when the environment is dangerous, increasing speed during a recurrent survey and extending the monitored area [13]. Robotic technologies that are used in a non-industrial environment to perform non-manufacturing activities are known as "service robotics" [5],[6]. According to ISO- 8373:2012 [16], a service robot refers to "*a type of robot that performs useful tasks for humans or equipment excluding industrial automation applications*". Operating in unconstrained environments, managing big data, requiring high intelligence for perception and small size for integration in the environment are the qualities that characterize the service robotics requirement domain [17].

The standard definition of service robot does not emphasize the aspect related to the autonomy [15]; however, according to Wirtz et al. [18] “service robots are system-based autonomous and adaptable interfaces that interact, communicate and deliver service to customers”. The capacity of perception, plan and execution (manipulating, navigating, collaborating) drive the autonomy of a robot in an environment [19]. Perception, planning and execution are linked with the possibility to collect data and extract information, which means combing the robotics (physical/hardware) [20] with artificial intelligence (AI) (virtual/software) for learning [10],[11],[19],[20]. Robots (ground, air, or water-based) or Uncrewed Aerial Vehicles (UAVs)[2] or Uncrewed Ground Vehicles (UGVs), with numerous sensors mounted on, such as Global Navigation Satellite System (GNSS)/ Global Positioning System (GPS), Light Detection and Ranging (LiDAR) and imaging system, collect the data. Artificial intelligence then converts the data in the information useful for ecosystems and environment observations[4],[11].

In the past, the employment of robots and AI approaches was limited by high-costs sensors and low-performance computers. Moreover, studies on these topics were also limited to very precise tasks for big companies or Public Authorities. Today, on the contrary, with the increasing the amount of cost-accessible robotic platforms and advances in the technology [23],[26], researchers are also exploring a range of new light-weight, spatial and spectral high-resolution sensors and reliable algorithms for automated data collection and interpretation.

1.1 Motivations

In the Age of Sustainability [27], [28], the primary role of artificial intelligence and robotics is linked to sustainable development and promotion of the United Nations Sustainable Development Goals (SDGs) at the global scale. Service robotics and AI, used as remote sensing (RS) technology, have been incorporated in various forms into the SDGs as technologies well-suited for environmental monitoring and urban socio-technical systems [29], [30], [31], [32]. Regardless of territorial scales, Service robotics and AI have to face challenges associated with rapidly growing urban populations, environmental degradation, pollution and climate changes.

[2] In literature there are different nomenclatures for ground or aerial “drones”. The most common is unmanned aerial vehicles (UAVs) or unmanned ground vehicles (UGVs). However, according to the study reported by Garr-Schultz and Gardner [24], the term “unmanned” (or conversely “manned”) holds strong gender connotations, due to the reference to a male space. Thus, as proposed by the National Aeronautics and Space Administration (NASA), the use of term “uncrewed” have been introduced as gender-inclusive terms.

The evolution of digital solutions generates new ideas in many domains. Technical innovation and ICTs (Information and Communication Technologies) offer a new opening for managing Urban and Regional Development (URD), providing new tools for urban planners. In the theory of URD, the holistic concept of “smart” indicates the embedding of the advances in technology and data-gathering into the environments where people live [33]. In this sense, the word “smart” directly calls upon the use of artificial intelligence, also referred with Geospatial Artificial Intelligence (GeoAI) [34], and robotics in cities and their services [35]. According to “triple helix of smart cities models” [36], cities are “regional innovation systems with knowledge central to growth and linked by technology and informatics” [37], [38].

The connection between service robotics, AI and URD introduces a new set of technologies that Nagenborg in [39] defines as *urban technologies*. *Urban technologies* do not represent an ontological concept, i.e. a concept with well-defined characteristics in a well-bounded domain but identify an open field of enquiry, a domain without pre-established definition. This class of technologies has a core function in promoting the idea of smart city, smart agriculture and farming according to the goals of the New Urban Agenda Habitat III [40], such as sustainability, safety and resilience. Therefore, the literature refers to “urban technologies” as the set of technologies that can have an impact on cities and its governance without specific common properties. Positional systems are the most developed examples; however, even monitoring systems could be included as soon as the technology becomes more advanced, miniaturized and mainstream. Indeed, in the monitoring system field, the remote sensing, the data analysis, and the interpretation tools need further enhancements to improve their use at different scales.

In this scenario, the activities of PIC4SeR (PoliTO Interdepartmental Centre for Service Robotics) well fit into the framework of “urban technologies”. PIC4SeR coordinates the activities of various research groups on technologies enabling service robotics [41]. Its purpose is to integrate innovative solutions in the areas of control, perception, AI with a focus on user interaction, security, socio-economic and ethical factors. The Centre selects different application areas: Smart Cities & Search & Rescue, Service robotics for wellbeing, Cultural heritage and Precision Agriculture. In these application fields, the Centre's idea is to develop of a multi-agent and multi-sensors platform, i.e. Uncrewed Aerial Systems (UASs) and Uncrewed Ground Systems (UGSs), that collaborate among themselves to allow various perspective and to overcome the limits of a single platform. Some of these shortcomings, for example, are limited capabilities of the batteries, restrictive flight regulations for the aerial vehicles and narrow UGV field of view.

Figure 1 shows a general overview of PIC4SeR's core concepts, for instance, in Precision Agriculture (PA) application.

This Ph Doctoral dissertation, funded by PIC4SeR and led by its purposes, is motivated by the idea to enhance *urban technologies* related to environment observation, monitoring and inspection. In this way, the work aims at filling the gaps for implementing SDGs in the "smart service" field.

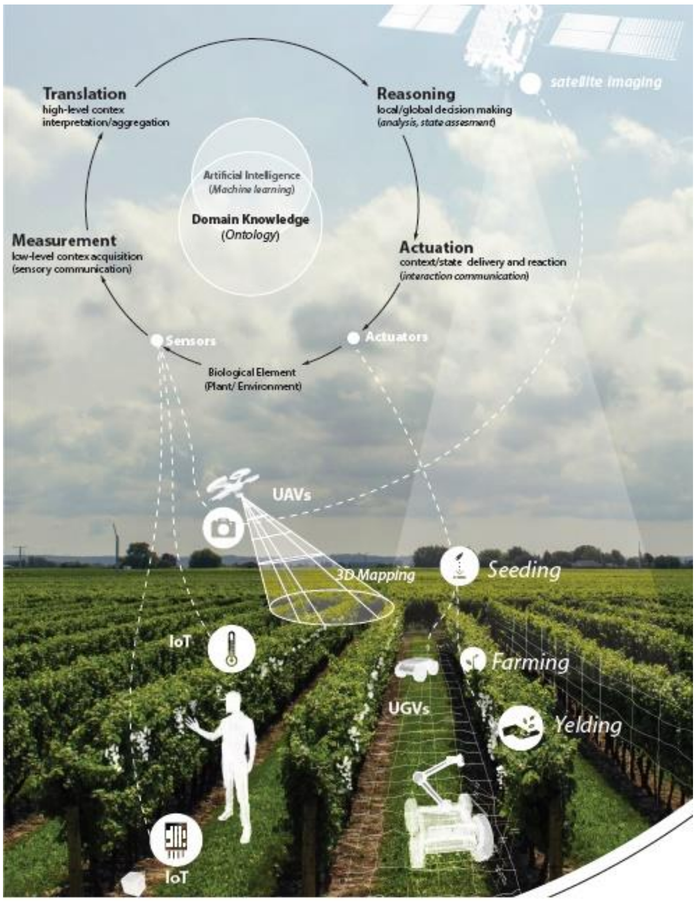


Figure 1. PIC4SeR concept idea: example of Precision Agriculture application [42].

1.2 Research Objectives

This thesis seeks to develop a quasi-real-time automated information extraction technique applied to high resolution, multi-scale, multitemporal and multi-sensory data for monitoring and for the inspection in application fields where object detection is particularly challenging. Among these last precision agriculture and aircraft inspection for de-icing operations are two relevant examples. The standard process to achieve the definition of an effective combination of technologies and techniques implies many domains such as robotics, remote sensing and computer vision. Therefore, the Information Extraction (IE) process passes through three main steps: (1) platform definition, (2) sensors configuration and calibration and (3) information extraction.

Hence, the present research runs through the whole process focusing on UAV platform configuration to integrate innovative sensors such as hyperspectral camera and, at the same time, testing different methodologies for extracting information from data characterized by different spectral and spatial resolution such as Red, Green, Blue (RGB), multispectral and hyperspectral images.

Numerous researchers have provided solutions able to face the problem of automatic information extraction from remote sensing and several studies reveal that the most innovative robotic systems equipped with high-resolution imagery sensors and Machine Learning (ML) techniques have been applied to large-scale for monitoring and mapping [1], [43], [44]. Although, reliable real-time automated methods in natural light and at different scale scenes are still missing and the ability to resurvey and detect objects, have significant gaps in the current state of the art. The limitations identified in the different monitoring application concerns:

- the collection and the delivery of information in a time-effective way;
- the lack of high spatial and temporal resolution;
- the need to have multiple points of view and level of details to improve the scene understanding;
- the complexity to deal with high spectral and spatial resolution data and with a large amount of data in the real-time application.

The research, facing with these central issues, suggests a solution analysing two application fields. Therefore, two study cases are chosen as examples of testing technologies and techniques. Investigating these cases and their singularities deeply, the thesis treats these three following aspects: (1) system configuration and calibration, (2) semantic segmentation using hyperspectral images, (3) object detection using RGB images.

1.2.1 System configuration and calibration

The first point addressed in this research relates to the optimal platform-sensor system configuration that can fit the detection and spectral characterisation of materials. It is known that the use of a spectral sensor can precisely define materials also out of the visible spectrum. Until now, material spectral characterisation has typically been measured in laboratories or in the field with handheld spectrometers that are in proximity of the target surface [45], [46], as it requires high spatial and spectral details. However, the introduction of UAV spectral remote sensing techniques leads to excellent advantages in monitoring and inspection applications, as they allow to analyse multiple targets in large areas with also high temporal resolution. Therefore, for what concerns the platform, the study focuses on the mainstream and more suitable platforms such as UAVs, because, despite the satellite or aerial platform, they can provide a higher spatial, spectral and temporal resolution. However, even though the platform identification seems rather simple, the sensor selection reveals to be complex. In the domain of spectral sensors, the multispectral and hyperspectral can be employed.

On one hand, the multispectral camera has a lower spectral resolution, but it is less expensive than the hyperspectral camera. On the other hand, the hyperspectral camera is a high-cost sensor, but it can provide the highest spectral resolution. Thanks to this feature, the hyperspectral camera can handle homogeneous surface materials' characterisation by a unique spectral signature. Nonetheless, since light-weight UAV hyperspectral cameras are not yet a deeply known technologies for close-range remote sensing, different aspects have not yet been solved, in relation e.g., with its integration with the drone, its calibration and band co-registration. Besides, techniques and dedicated software to solving these problems are still under development.

For these reasons, this work explores the complex use of hyperspectral sensors and investigates the possibility of employing multispectral cameras, as “low-cost” sensors. The chosen application field to test these issues is the ice detection for aeroplane inspection and de-icing operations using a UAV platform. The characterisation of the ice, i.e. a material that is invisible and can change its physical state, is challenging and can be achieved through very powerful spectral sensors. Thus, first, the more advanced hyperspectral camera has to be selected. Then, the integration with the UAV platform is investigated, establishing, through a deep research among the products on the market, all the hardware components required. In the end, the system calibration and the data co-registration are investigated implementing specific routines for data correction. In addition, a comparative analysis between these two sensors has been performed analysing the advantages and the drawbacks of multispectral and hyperspectral sensors and studying the impact of the different spectral resolutions on the detection accuracy.

1.2.2 Semantic segmentation using hyperspectral images

Besides the hardware for the data acquisition, for the detection and the characterisation of materials and their feature, an information extraction process that involves hyperspectral or multispectral data is required. This means dealing with high spatial and spectral resolution data, that are difficult to manage and requires a time-consuming procedure for the analysis. Thus, big data management and definition of ML quasi-real-time algorithms are the most crucial aspects deeply addressed in the present research, adopting the ice detection application as a test case.

Considering the hyperspectral data exploitation problems related to the big data management, the distinction among spectra of the same material, spectral signatures of different materials, and the significative parts of the spectrum describing a material are analysed. As pointed out in this study, big data management can be addressed in two different ways. The first solution consists of the dimensionality reduction in post-processing, while the second is the reduction of spectral resolution during the data collection phase. Both solutions involve the use of an ML algorithm able to extract and select the significative features. For this reason, a Principal Component Analysis (PCA) is tested. In this way, the identification of the significative part of spectral response for the studied materials, and thus the distinctiveness among spectra, is also addressed. It is worth to underline that the significative band selection and the definition of distinctiveness among spectra are tricky problems. Indeed, in real-world hyperspectral imagery, using remote sensing techniques, various effects can induce a variability to the material spectra [47]: atmospheric conditions, sensor noise, material composition, scene geometry, Ground Sample Distance (GSD), and surrounding materials are some examples. This variability prevents the identification of spectral signatures. The term spectral signature, in remote sensing, is often misleading as it is not a unique line for each material, and different materials can have almost identical reflectance spectra. To overcome these issues, indoor and outdoor datasets have been analysed.

Thanks to PCA outcomes, the reduction of the spectral resolution during the data acquisition can be addressed using a multispectral camera that covers the same spectral range or hyperspectral camera with only the representative channels. In the end, two ML algorithms are tested, the Random Forest (RF) [48] and the Support Vector Machine (SVM) [49], to choose the best solution in terms of accuracy and computing time. The approach shows that machine learning algorithms can process high spectral resolution data, and different solutions can be adopted for real-time data processing.

1.2.3 From semantic segmentation to object detection using RGB images

It is known that UAV platform use also has the advantage of allowing the collection of multi-view, multi-sensor, and multi-scale data. As in spectral image processing, the analysis of such multimodal datasets can be instrumental in monitoring applications, because it can simplify the interpretation process. However, using these types of data in combination with automatic information extraction techniques requires further investigations. Indeed, the automatic real-time algorithm has to manage very different types of data. To explore this aspect for real-time analysis, UAV RGB original images collected in real conditions, with different points of view and various sensors are employed. In this case, a precision agriculture-related application such as plant disease detection has been chosen as representative. The challenge is to distinguish the object and its property (e.g., the vegetation status) at UAV-scale. In general, data collected in a single flight with a nadiral pattern have been employed [47],[48], while, in this case, the use of multi-sensory, multi-view, multi-resolution, and multi-temporal data have been tested at the same time.

The introduction of such types of data, also, prevent the necessity of the data augmentation for a better model generalisation and at the same time allows to transfer the learning process to different datasets. Indeed, the data augmentation can be tricky in precision agriculture application. Data augmentation methods are exploiting the shape of objects applying (1) rotations or flip, (2) crop or (3) colour modifications [48],[49]. In the studied case, where the disease affects only the radiometric response and not the plants' shape, the listed data augmentation methods are not efficient.

The more varied the data and the more generalized the model, the higher is the possibility of using the same algorithms for information extraction process on new datasets, avoiding the training process. The training process is, indeed the most time-consuming part in detection problem. Thus, a generalised model means having a pretrained model that can significantly enhance real-time applications. Thus, an approach to improve the model generalisation and to transfer learning has been studied and implemented.

Besides using different data types, in this application, two techniques are tested, moving from a semantic segmentation that is a pixel-wise technique to an object detection method, based on the object. In real-time, it could be useful to detect only the object of interest (using object detection techniques) instead of segmenting the whole image (semantic segmentation). For this reason, for testing these two techniques, two ML algorithms that can deal with multi-sensory and multi-resolution data have been implemented in Python. The programming language has been chosen to further implement the same algorithms and integrate them into real-time hardware/software systems. The study focuses on Random Forest [48], and Faster Region-Based Convolutional Neural Network (Faster R-CNN) [54] as they could fulfil the needs of big data management in quasi-real-time. The approaches demonstrate that machine learning algorithms can rapidly ingest and process high resolution, multi-sensory, and multitemporal data.

1.3 Thesis’ contribution

Considering the open problems related to each case study described in 1.2, the main contributions of the thesis are:

- to give a whole overview of the current technologies and information extraction methods used for monitoring and inspection purposes;
- to define systems requirements and select the core technologies based on the application;
- to apply machine learning methods to quasi-real time automatic information extraction from multi-scale, multi-view and multi-sensory (RGB, multispectral, and hyperspectral sensors) in challenging applications and test transfer learning;

The work presented in this Ph.D. thesis has been published in some peer-reviewed journal papers and in some proceedings in different national and international conferences.

1.4 Structure of the thesis

The overall structure of the study takes the form of eight chapters. In details, Chapter 2 define the fundamental concepts of remote sensing and the state of the art of the platforms, sensors. Chapter 3 concerns the system configuration, the definition and the calibration of the sensors used in the selected study cases. Furthermore, due to the hyperspectral sensor complexity and the open problems related to its calibration, an in-depth description of the sensors selected for the thesis application, is presented. Chapter 4 provide a background and literature review of artificial intelligence algorithms for remote sensing applications. Then, Chapter 5 and Chapter 6 propose the methodology, the analysis and the results achieved in the two case studies as described in section 1.2. Chapter 5 presents the semantic segmentation using hyperspectral and multispectral images for ice detection in de-icing operations. Chapter 6 analyses both semantic segmentation and object detection using RGB multi-scale and multi-sensory approach in a precision agriculture application such as the plant disease detection. In the end, Chapter 7 provides conclusions and recommendations for future research.

Chapter 2

Remote sensing: physical concepts, sensors and sensor characterization.

Remote sensing is a broad term that refers to approaches for environmental study. It allows contactless acquisition of data for qualitative and quantitative characterization of surfaces and materials. Remote sensing can be a good alternative to the traditional fieldwork campaign thanks to the spatial and temporal coverage that characterize the produced data. Indeed, remote sensing offers the possibility to provide a consistent and repeatable methodology, suited equally to both quick as well as long term monitoring campaigns at several scales (from local to worldwide) [55].

The spatial, spectral and temporal features of information accessible from remote sensing, depends on specific properties of the sensors and platforms. Furthermore, according to the sensors/platform features, the illumination and the atmospheric conditions can affect data acquisition. Therefore, to be able to information extraction, the different artefacts inferred by the measurement conditions and sensor characteristics must be considered. Indeed, before the IE process, a fundamental part includes the characterization of sensors. It is worth to underline that, in general, the sensors are characterized, and data are calibrated. However, calibration is used to indicate both the two activities in remote sensing [56]. After the data correction, the information can be extracted to formulate interpretative hypotheses on the status and changes of objects.

This chapter sets forth the fundamental concepts of remote sensing (section 2.1), and presents a literature review of the leading sensing platforms (section 2.2), sensors (section 2.3) and methodologies for sensor characterization (section 2.4).

2.1 Overview of electromagnetic Remote Sensing process

As mentioned above, proximal remote sensing allows information acquisition about objects or scene with no direct contact, exploiting electromagnetic energy-matter interactions [1]. The main principle is that all objects with a temperature above absolute zero emit electromagnetic radiation. The detection of electromagnetic radiation involves four main components: a light source, atmospheric interaction, Earth's surface interaction, and sensors mounted on a platform[58].

The light source can be natural or human-made. Examples of a natural source can be the sun that illuminates the Earth or object with their self-emission due to their temperature. Instead, human-made sources can be microwave radar. Based on the source type, it is possible to distinguish active systems that emit the radiation and analyze the returned signal and the passive systems that acquire the sun radiation or the object thermal radiation. The electromagnetic radiation passes through the atmosphere and interacts with it. This interaction produces distortions on electromagnetic radiation. The main mechanisms causing modification or distortion of the radiation are scattering and absorption. The first generates variable radiation diffusion in the atmosphere. The atmospheric absorption induces an energy loss because the radiation is transformed. Regions with low atmospheric absorption characterize the electromagnetic spectrum. These regions are the atmospheric windows where remote sensing takes place [56], [59]–[61].

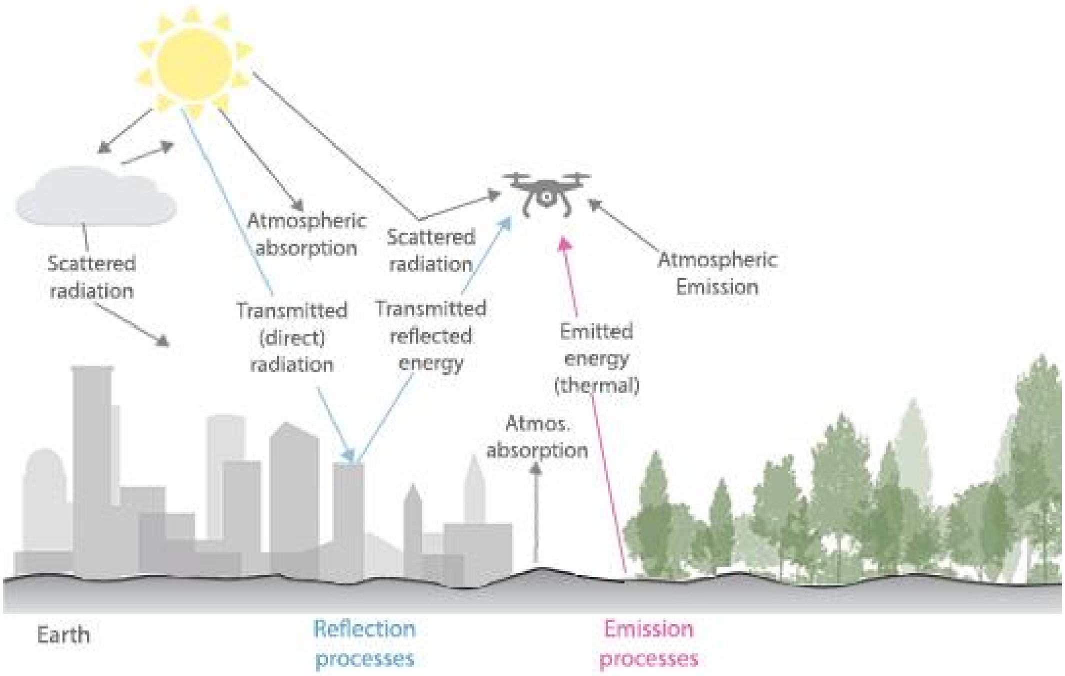


Figure 2. Energy interactions in the atmosphere and at the Earth's surface.

The Earth’s surface interaction results split in different fractions of the electromagnetic radiation. The nature of this interaction depends on the physical characteristics of the material and the involved energy. The energy indeed is reflected, absorbed, or transmitted (Figure 3). In remote sensing, the reflected part of the energy is the most important. It is possible to distinguish two types of reflection: specular reflection and diffuse reflection. The specular reflection occurs in the presence of a smooth surface when the energy is led away from the surface in a specific direction. The diffuse reflection is proper of rough surfaces (or Lambertian reflectors). In this case, the energy is reflected almost uniformly in all directions. In most of the case, standard materials are diffuse reflectors, although there are exceptions such as water.

The sensors, as an electro-optical detector, convert into an electrical signal the electromagnetic radiation detected as a compound of reflected solar radiation and emitted radiation by an object. It means that it is possible to acquire the spectral response of an item.

Unless the target emits the sensed energy, as described above, one of the main elements in remote sensing is the energy source that illuminates the target. The electromagnetic energy is propagated in the form of waves through electromagnetic radiation. The main features of electromagnetic waves are wavelengths (λ), energy (E), frequency (f). The electromagnetic spectrum refers to the whole range of electromagnetic waves. In Figure 3 the conventional representation of the spectrum. The radio, microwave, infrared, visible, ultraviolet, x-ray and gamma-ray radiation compose the entire spectral range [57].

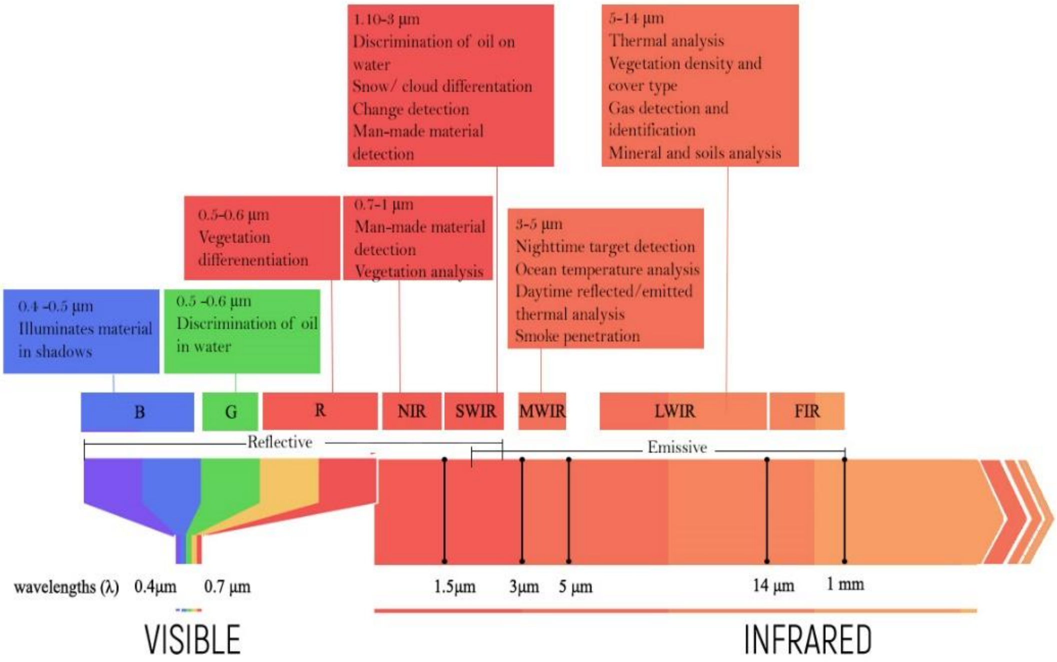

1.10-3 µm
Discrimination of oil on water
Snow/ cloud differentation
Change detection
Man-made material detection
5-14 µm
Thermal analysis
Vegetation density and cover type
Gas detection and identification
Mineral and soils analysis
0.5-0.6 µm
Vegetation differenentiation
0.7-1 µm
Man-made material detection
Vegetation analysis
3-5 µm
Nighttime target detection
Ocean temperature analysis
Daytime reflected/emitted thermal analysis
Smoke penetration
0.4 -0.5 µm
Illuminates material in shadows
0.5 -0.6 µm
Discrimination of oil in water
B
G
R
NIR
SWIR
MWIR
LWIR
FIR
Reflective
Emissive
wavelengths (λ)
0.4µm
0.7 µm
1.5µm
3µm
5 µm
14 µm
1 mm
VISIBLE
INFRARED

Figure 3. Electromagnetic spectrum and spectral information present at different spectral regions. NIR refers to Near- infrared, SWIR to Short Wavelength Infrared, MIR refers to Medium wavelength Infrared and LWIR to Long Wavelength Infrared, and FIR to Far Infrared.

2.2 Sensing Platform

The sensors must place on a stable platform to collect and record the data. It is possible to distinguish sensing platforms in the ground platforms (UGS, and static station), aerial (aircraft and UAS) and space platforms (space shuttle or satellite) [62]. Another possible classification of the platforms is in remote or proximal sensing according to the distance from the target.

Remote sensing platforms can be considered: satellite, airborne and LARS (Low Altitude Remote Sensing) system [63]. Instead, proximal sensing is performed using sensors near the surface of the Earth [64]. In this case, the sensors are mounted either on common vehicles based on the application such as cars and, agricultural vehicles, vehicles specifically designed for in-the-field monitoring [65] or uncrewed vehicles.

As mentioned above, the choice among the use of these platforms is related to many parameters as reported by Toth et al. [66],[62], [67]. The attention is focused on the parameters below:

- observation space and ground coverage;
- spatial resolution, expressed in GSD; the GSD is the smallest area viewed by the sensors and can be represented in the digital image;
- spatial accuracy;
- data acquisition frequency or revisit time.

According to the selected parameters above, Table 1 lists the most frequently platforms and their features.

Table 1. General specifications of typical platforms.

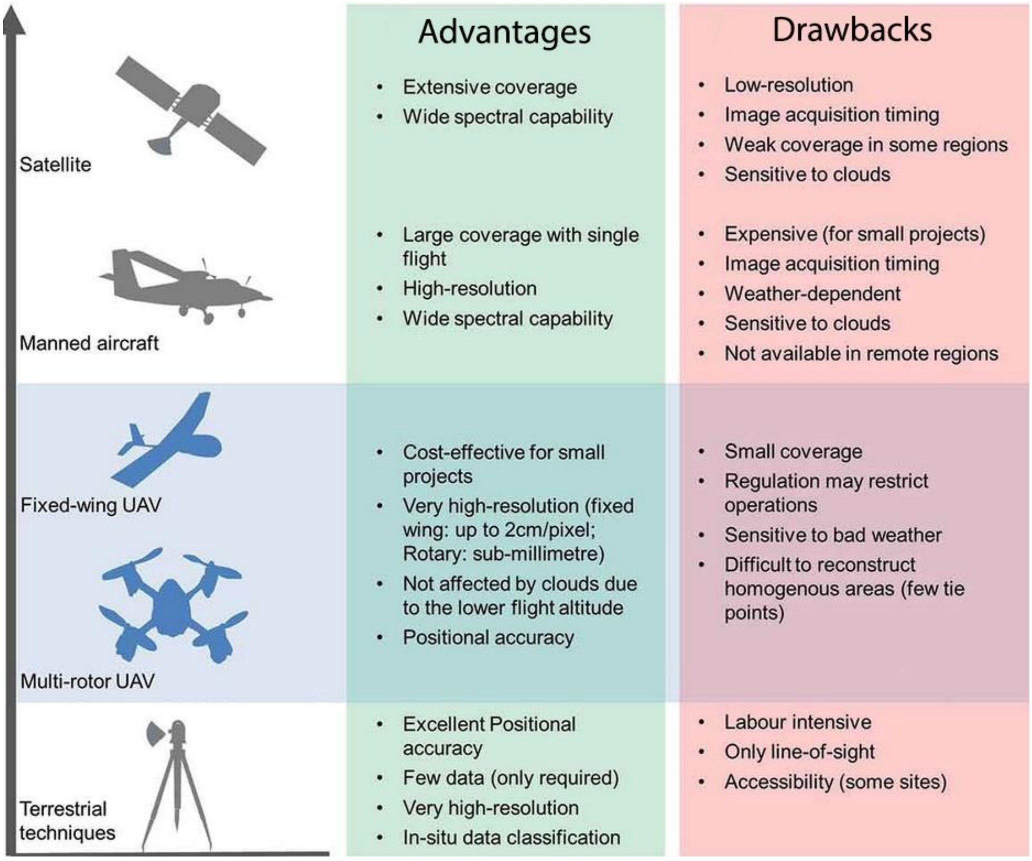
Platform	Covered surface	Spatial resolution	Spatial accuracy	Revisit time
Satellite	Global/National (100 km^2)	0.30–300 m	1–3 m	days
Airborne	Regional (10 km^2)	5–25 cm	5–10 cm	hours
LARS system	Local (1 km^2)	1–5 cm	1–25 cm	minutes
Ground station	Site (100 m^2)	5mm–5 cm	5 mm–50 cm	minutes

As Table 1 shows, conventional satellite-based remote sensing has a significant limitation to get data at resolution from 30 m to 300 m, which is coarse for many applications. There are few examples of commercial satellites (e.g. GeoEye-1) that provide sub-meter resolution imagery (spatial resolution < 1 m for panchromatic and > 1 m for multispectral); nevertheless the data are referred to a given time and place/area with a long revisit time at a given price. Moreover, the commercial satellite takes usually seven days to provide processed images. The satellite images may be the right solution for the case of extensive areas; however, weather conditions can also be a limit in this case.

Airborne remote sensing used to be the primary source of geospatial data for the developed part of the world until the introduction of commercial satellite systems about a decade ago. The high cost and the revisit time, instead, are also the main drawback of the airborne platforms as well as space-borne systems. For many environmental applications, the conventional airborne and satellite platforms require trade-offs in terms of high cost, operational flexibility, limited versatility, and (or) low spatial and temporal resolution [68]. Consequently, images taken by low-altitude remote sensing platforms such as uncrewed aerial vehicles or uncrewed ground vehicles provide an alternative [69].

Over the past few decades, UAV of various shapes, sizes, and capabilities have been developed. The interests in their potentials for civilian applications are overwhelming, even though in many countries, the regulation of UAV use into the national airspace is still work in progress [70]. Indeed, compared with the conventional satellite and aircraft platforms, several characteristics make uncrewed aerial vehicles highly attractive to remote sensing, including their: (1) low cost; (2) ability to perform missions and acquire data autonomously with minimized human interaction; (3) manoeuvrability, which is ideal for low-altitude flying and navigating complex environments; (4) ability to operate in adverse weather and dangerous environments; (5) reduced exposure risk to pilots.

Figure 4 summarizes the advantages and drawbacks of each type of platform.



	Advantages	Drawbacks
Satellite	• Extensive coverage • Wide spectral capability	• Low-resolution • Image acquisition timing • Weak coverage in some regions • Sensitive to clouds
Manned aircraft	• Large coverage with single flight • High-resolution • Wide spectral capability	• Expensive (for small projects) • Image acquisition timing • Weather-dependent • Sensitive to clouds • Not available in remote regions
Fixed-wing UAV Multi-rotor UAV	• Cost-effective for small projects • Very high-resolution (fixed wing: up to 2cm/pixel; Rotary: sub-millimetre) • Not affected by clouds due to the lower flight altitude • Positional accuracy	• Small coverage • Regulation may restrict operations • Sensitive to bad weather • Difficult to reconstruct homogenous areas (few tie points)
Terrestrial techniques	• Excellent Positional accuracy • Few data (only required) • Very high-resolution • In-situ data classification	• Labour intensive • Only line-of-sight • Accessibility (some sites)

Figure 4. Advantages and drawbacks of the remote-sensing platforms [71], [72].

Several papers and reviews proposed different taxonomies for classifying UAVs [71], [73], [74],[75], [76] and pointed out their main features (e.g. weight, flying altitude, payload, endurance, and range) [67], [77], [78]. However, two major types of drones can be identified: fixed-wing, or rotary-wing [71]. The fixed-wing drones are faster and preferable for covering large area, while the rotary-wing drones may be preferable for achieving high spatial resolution measurements and carrying more than one sensor. Despite being highly flexible platforms, the limitations of multi-rotor systems are their endurance (<30 minutes) and speed, restricting them to a relatively small area (<5 km^2). UAV platforms were not originally developed for remote sensing; thus, the technology is mainly coming from the large consumer market. Since our UAV-based remote sensing platform is to be transportable and to be operated locally under minimal legal restrictions, the common solutions are concerned with the small systems, defined by a maximum total weight of 25 kg [62]. Even if the official classification is proposed by van Blijenburgh [79], for these research activities, Table 2 summarizes the small UAVs available employed for remote sensing applications, according to the classification based on the weight of the UAVs proposed by Thenkabail et al. in [80].

Table 2. Types of drones and overview of the representative examples on the market.

		Payload	Maximum flying time	Products
Small size UAV (<2g)	Fixed wing	<2 kg	> 1 hour	SenseFly eBee, RTD X5, Parrot Disco, Lehmann Aviation LA500, and Baaz Flying Wing
	multi-rotor	< 1 kg	<30 minutes	DJI Phantom® 4, 3DR Solo Quadcopter®, Topcon Falcon 8®, SenseFly eXom®, Yuneec Typhoon 4K®
Large size UAV (>2 kg)	Fixed wing	>1 kg	>30 minutes	QuestUAV Q-Pod, Trimble UX5, Precision Hawk Lancaster, Penguin B, Sentera PHX
	multi-rotor	0.5 < p < 9 kg	>30 minutes	MicroKopter 3500-Geo®, Yuneec H920 Tornado®, DJI Matrice 210®, DJI Matrice 600®, and Freefly Systems Alta 8® [79], AgBot [81]

Despite the several ready-to-use solutions for remote sensing available on the market for what concern the UAVs, the UGVs did not reach yet the same degree of evolution. Indeed, they have the drawback to be constrained to two-dimensional surfaces. In the last years, few attempts of developed cooperative platforms such as UGV platforms combined with UAVs to fill the gap of the aerial view were made [5], [82].

The features of the platforms carrying the remote sensors play a significant role in the efficiency of the object observation. The more evenly the observation space, the higher the observability. There are several approaches to improve the observing potential from a platform. In addition to the cooperative sensing promoted using multiple platforms, as cited above, the current trend is using multiple sensors in different orientations on the same platform. This approach represents the current trend of multi-sensory systems. Thus, regardless of the type of platform, a single system can mount several kinds of sensors. The first remote sensing systems were single-sensor platforms. However, since sensors have become more affordable with improvement in sensing and computer technologies, the modern RS systems use multiple sensors, including identical, and different sensors, such as numerous cameras [64]. The following section presents the main types of sensors.

2.3 Remote sensing sensors for UAV

Remote sensors can be classified in active and passive sensors. The active sensors provide the own energy source for illumination such as LiDAR, microwave (Radio Detection And Ranging - RADAR), or Sound Navigation and Ranging (SONAR) [66]. The passive sensors can measure the naturally available energy. Standard RGB cameras, hyperspectral, multispectral, thermal camera sensors are some examples. In this review, passive sensors will be the focus.

Imaging sensors store data as a matrix of picture elements called pixels. Each pixel records a representative number of the brightness or intensity level of the radiant energy. The image data characteristics depend on the sensor’s resolutions: spatial, radiometric and spectral [80], [83]. The spatial resolution depends on ground size of the pixel and it is expressed in m. It influences the resolution of the scene details and it is related to the platform altitude and the instantaneous field of view (IFOV). IFOV is the minimum solid angle through which radiant energy that reaches a single detector element. The radiometric resolution, also called radiometric sensitivity, describes the sensor sensitivity to discriminate very slightly different intensity of energy [83]. It concerns the number of divisions of bit depth (e.g., 255 for 8-bit) in data collected by a sensor.

The spectral resolution is the ability of the sensor to define fine wavelength intervals for each spatial sample [83] [84]. Therefore, higher is the spectral resolution, better is ability of the sensor to exploit differences in spectral signature. The spectral resolution indicates the number of bands that the sensor can acquire and the bandwidth [80]. Thus, the spectral resolution limits the number of bands that the sensor can acquire. Depending on spectral resolution, the classification of optical sensors includes:

- panchromatic (PAN) imaging sensor: single-channel detector within a broad wavelength range;
- RGB imaging sensor: three bands detector in the visible domain;
- multispectral imaging sensor (MSI): can acquire few bands (from 2-9 bands) within a narrow wavelength band. The most operational sensors capture images in the visible to NIR (VNIR) domain;

- hyperspectral imaging sensor (HSI): can collect more than 10 continuous narrow (10-20 nm) spectral bands in the spectral range from 0.4 to 2.5 µm. However, it is worth to underline that some authors, [83], [85], report a further discrimination among superspectral and hyperspectral sensors, where the superspectral sensors can collect a limited continuous spectral bands from 10-16, instead HSI can acquire more than 16;
- thermal imaging sensor: single-band detector that can operate in the spectrum at wavelengths from 5 µm to 14 µm;
- SWIR imaging sensors: can acquire bands in the spectrum at wavelengths from 1.5 µm to 3 µm [86], [87].

Table 3 outlines the main difference among RGB, multispectral and hyperspectral according to spectral and spatial resolution [88].

Table 3. Difference among imaging sensors. A bullet rate (1-4) was used to quantify both spectral and spatial information associated with each sensor. One refers to the lowest resolution and four to the higher resolution.

Sensors	Spatial resolution	Spectral resolution
PAN or RGB imaging sensor	****	**
Multispectral imaging sensor	***	**
Hyperspectral imaging sensor	**	****
Thermal imaging sensor	*	*
SWIR imaging sensors	**	*

The relation between spatial and spectral resolution is related to the format of a detector. It is possible to reduce the size of the detector element, thus increasing the spectral resolution, only if it is possible to increase the received energy for a wider spectral band. The hyperspectral sensors represent a trade-off in which spatial resolution is reduced in favour of the spectral resolution, and they can handle applications in which the spectral information is more reliable or measurable than the morphology or shape information. [79] carried out an in-depth survey on optical remote sensing instruments for uncrewed vehicles. The successive sections concentrate on sensors which may be suitable for UAV or UGV platforms, distinguishing among RGB, multispectral, hyperspectral sensors (Figure 5).

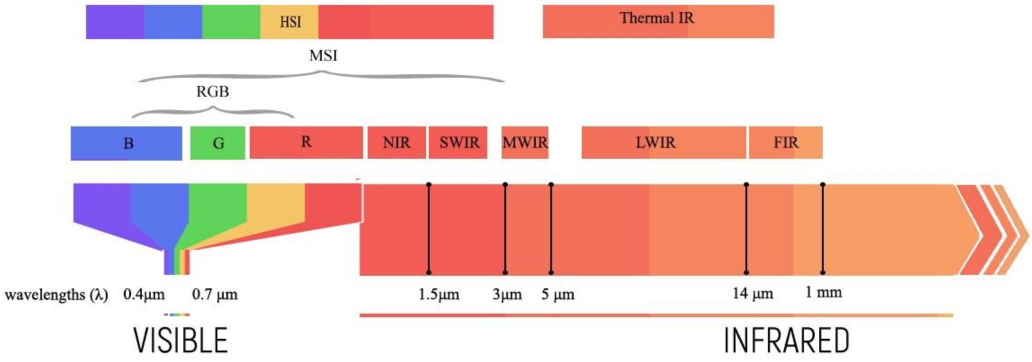


Figure 5. RGB, Multispectral, and hyperspectral sensors and the electromagnetic spectrum.

2.3.1 RGB sensors

The RGB sensors take information in three bands red, green, and blue. These cameras are designed to capture data from a visible portion of the electromagnetic spectrum [89]. Each channel can be considered as the signal strength within the selected wavelengths: red (650-750 nm), green (495-570 nm), blue (450-495 nm).

Compared to the sensors described in the section 2.3.2 and 2.3.3, there are a wide range of RGB cameras available on the market. The RGB camera can be made up of two types image chips: charge-coupled device (CCD) or complementary metal oxide semiconductor (CMOS) chips (pixel size and noise level). The standard parameters for selecting an RGB camera are camera lens, resolution, the chip quality and the shutter type [67]. They usually are highly integrated on platforms; they are miniaturized and have affordable costs.

For remote sensing application, for the interpretation of images, it is important knowing the sensor spectral response curve, which is a function of the wavelength of signal relative to detection of color. The sensor response functions for consumer-grade cameras have a similar shape to the spectral sensitivities of human cone cells, since the aim of these products is to capture a representation of the scene that is as accurate as possible from the point of view of human perception [90]. Figure 6 reports as example sensitivity curve for Hasselblad X1D camera [91], [92].

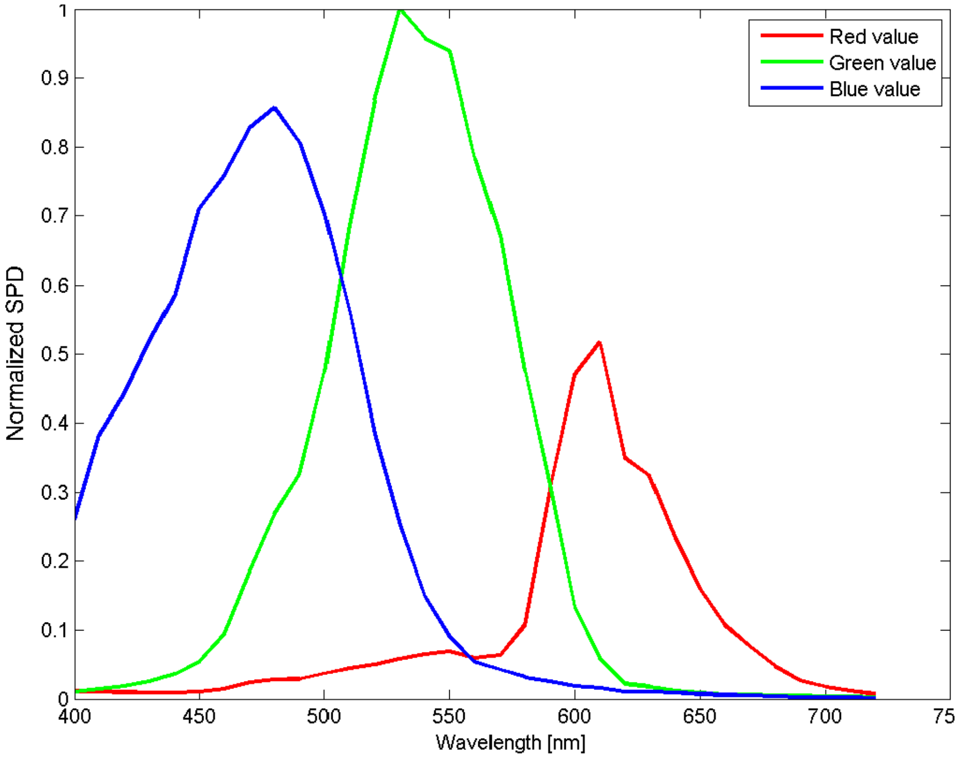


Figure 6. Spectral sensitivity functions for Hasselblad X1D camera.

A review of the main RGB sensors used for RS application is presented by in Colomina and Molina [70]. Based on the camera resolution it is possible to distinguish among small-format (SF) sensors (up to 16 megapixels sensors), medium-format (MF) sensors (between 16 and 50 megapixels) and large-format (LF) cameras (with more than 50 megapixels sensors). The distinction was initially based on the size of the sensor. In the past, the size of small-format sensor is 24×36 mm, medium-format sensors have a size up to 60×90 mm, and everything bigger was large. Table 4 reports an updated overview of the most common and representative medium format and small format RGB sensors solutions now available on the market [93].

In addition to the most common cameras, some developments consisting of multiple-head RGB cameras have been performed, and results have been recently published. Some example can be the four single cameras Canon EOS 5D Mark II, five cameras such as Crevis MV-CS27U USB. Moreover, the development of smartphone equipped with high-quality RGB cameras leads to experimenting with this technology as a UAV payload [94], [95].

Table 4. Overview of common and representative SF and MF RGB sensors on the market and their specifications. IN refers to In-built lens and C to changeable.

Products	GoPro Hero4 Silver	Ricoh GR2	Flir Blackfly S BFS-U3 series	Sony A7 series	DJI Phantom PRO	Hasselblad X1D	Phase One iXU 1000
Format	SF	SF	SF	MF	MF	MF	MF
Shutter	Rolling	Global	Global	Rolling	Rolling	Rolling	Rolling
Sensor type	CCD	CMOS	CMOS	CMOS	CMOS	CMOS	CCD
Sensor size	5.9×4.5 mm	23.7×15.6 mm	14.57×4.38 mm	35.8×23.9 mm	13.2×8.8 mm	43.8×32.9 mm	53.4×40 mm
Pixel size	1.5 μm	4.8 μm	2.74 μm	5.97 μm	2.41 μm	5.48 μm	4.60 μm
Focal Length	5 mm	18.3 mm	[-]	[-]	24 mm	[-]	[-]
Lens optics	IN	IN	C	C	IN	C	C
Pixel Number	12 MP	16 MP	24.5 MP	24.3 MP	20 MP	50 MP	100 MP
Image size	4000×3000 pixel	4928×3264 pixel	5320×1600 pixel	6000×4000 pixel	5472×3648 pixel	8272×6000 pixel	11608×8708 pixel
Weight	83 g	220 g	280 g	769 g	[-]	1200 g with lens	1600 g

2.3.2 Multispectral sensors

Multispectral sensors are designed to record more than the three primaries. The number of bands is typically from 3 to 10. Each channel is sensitive to radiation within a narrow wavelength band. The result is a multilayer image that contains both the spectral information and brightness of the observed targets [76, 77]. The multispectral camera can be divided into two categories based on bandwidth:

narrowband and broadband [98]. Narrowband refers to a camera with a small spectrum range (e.g. Micasense has a bandwidth of 10 nm) [99]. Small spectral range, generally, corresponds to a more accurate spectral measurement. Broadband means wide spectral ranges cameras (e.g. MAPIR Survey3N has a bandwidth of 45 nm), which are usually like those of satellites. The use of these types of camera simplifies the application of algorithms used for satellite-based sensors [100]. However, according to Zhao et al. [98] and Deng et al. [101], the results derived from narrowband and broadband cameras are quite similar.

As in the case of RGB camera, for the interpretation of images, the sensor response functions are important. Figure 7 reports as example spectral response of Parrot Sequoia[102].

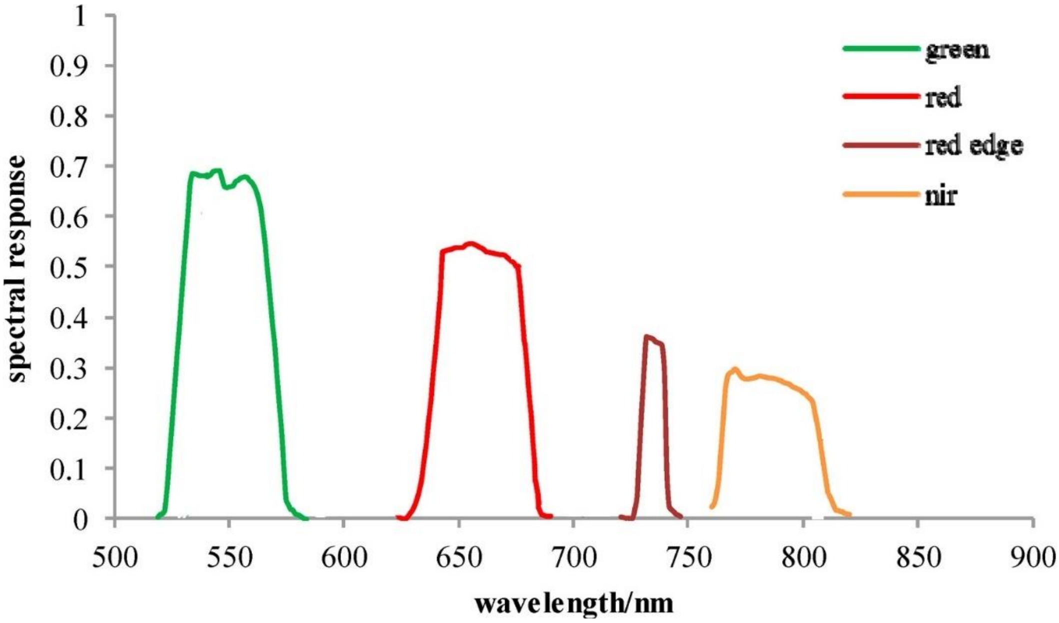


Figure 7. Spectral response functions for Parrot Sequoia camera.

A critical factor for MSI camera is the radiometric calibration for determining spectral accuracy and reflectance-based derivatives. Before using multispectral imagery, radiometric calibration is a prerequisite and an essential step, in which the digital number (DN) value recorded by the camera can be converted into the spectral reflectance. Some MSIs allow pre-flight calibration, while others require a vicarious procedure (the section 2.4.3 reports details regarding radiometric calibration). In general, a narrowband MSI that uses vicarious calibration, instead of broadband camera that uses pre-flight calibration. Furthermore, thanks to its non-scanning technologies, it is possible to estimate three-dimensional (3-D) scenes based on the structure from motion (SfM) technique.

Regardless of the reported characteristics, and compared with RGB cameras, MSIs are more expensive because of extra hardware for handling the additional bands. Moreover, due to limited application, the data format compatibilities and the software packages are under development. Table 5 summarizes commercial multispectral sensors used in RS application and the specification [70], [103], [104].

Table 5. Overview of main commercial multispectral sensors and their specifications. HFOV stand for the horizontal field of view, VFOV refers to the vertical field of view.

Products	MAPIR Survey3N	Slantrange 4P+	Tetracam ADC Micro	Micasense Altum	Parrot Sequoia	Sentera 6x	DJI P4 Multispectral
Shutter type	Rolling	Global	Global	Global	Global	Global	Global
Sensor size	6.20×4.65 mm	7.04×5.28 mm	6.55×4.92 mm	7.12×5.33 mm	4.8×3.6 mm	7.11×5.33 mm	4.96×3.72 mm
Pixel size	1.55 µm	3.44 µm	3.2 µm	3.45 µm	3.75 µm	3.45 µm	[-]
Image size	4000×3000 pixel	2048×1536 pixel	2048×1536 pixel	2064×1544 pixel	1280×960 pixel	2048×1536 pixel	
Focal Length	8.25 mm	8.00 mm	8.43 mm	8.00 mm	3.98 mm	8.00 mm	5.74 mm
Lens optics	41° HFOV	[-]	42.48° HFOV, 32.5° VFOV	48º HFOV, 37º VFOV	61.9° HFOV, 48.5° VFOV	47° HFOV	62.7° HFOV
Spectral range	385-950 nm	410-950 nm	520-920 nm	475- 840 nm	550-790 nm	475-840 nm	450-840 nm
Bandwidth	[-]	[-]	[-]	10 nm	[-]	[-]	16 nm
Pixel Number	12 MP	[-]	3.2 MP	3.2 MP	1.2 MP	3.15 MP	2.8 MP
Spectral channels	4	6	3	5	4	6	5

2.3.3 Light-weight hyperspectral sensors

Hyperspectral sensors are known as imaging spectrometers [80]. This sensor collects simultaneously digital images in many relatively narrow, contiguous and non-contiguous spectral bands of the electromagnetic spectrum [57]. The imaging spectrometers are integrated systems composed of the fore optics, the spectrometer, and the detector or detector array. The spectrometer can split the radiance into several wavelength bins either through dispersive or interferometric means. The focal plane array collects the signal as digital numbers.

The dispersive spectrometers can use prisms for diffractive elements to separate the light into the wavelength bands or grating that spatially divides the wave-front, introducing a path difference. Interference spectrometers use amplitude division of the wave-front to extract the spectral information. Two examples of interference spectrometer are Fabry–Pérot (more details are presented in section 3.2.1) and Michelson interferometers.

The data are stored in a data structure known as the data cube. It is possible to distinguish different acquisition method of imaging spectrometer for storing data. Indeed, according to the way in which the data cube is populated is possible to identify three main types of hyperspectral sensors: whiskbroom that utilizing a rotating mirror that scans the projected slit across the surface with the long axis parallel to the flight direction, push-broom that scan a single line, and frame-based (staring) that captures the whole image (Figure 8) [105].

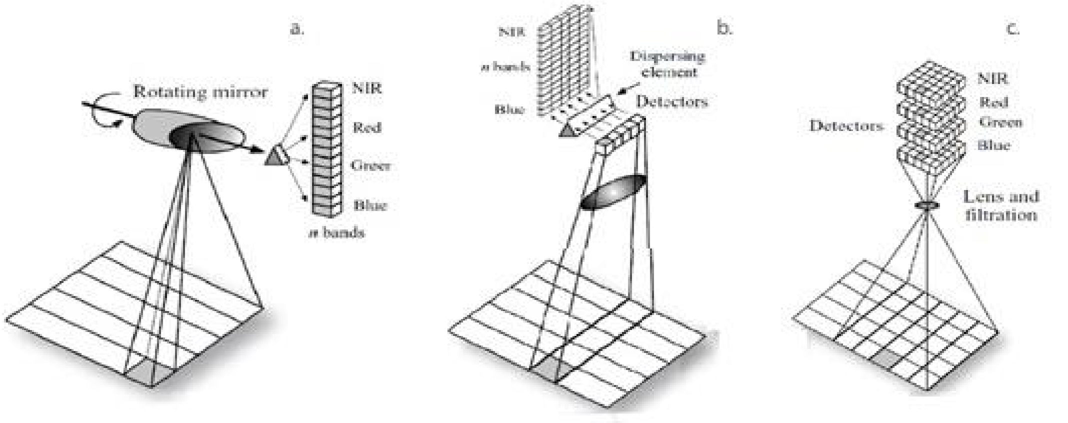


Figure 8. Types of hyperspectral sensors: (a) whiskbroom, (b) push-broom sensors, (c) full-frame sensors [66].

The whiskbroom and push-broom are usually dispersive systems. Instead, the staring system employs interferometer. Push-broom sensors were the most frequently used imaging technique as hyperspectral solutions, and it is also the mainstream imaging method for VNIR and SWIR imagers. The advantage of push-broom sensors, compared with the whiskbroom, is that the spectral information of all the points along a line is measured simultaneously. Nonetheless, to reconstruct an accurate cube, a push-broom sensor usually requires a stabilized UAV platform that allows it to move uniformly in a straight line and a GNSS and Inertial Measurement Unit (IMU) module to register accurate attitude and position information for the postprocessing. In UAV-borne applications, the accuracy of a push-broom hyperspectral cube is a function of the sensor frame rate, relative flight speed, and attitude and position information of the sensor. Compared with the push-broom sensors, the core advantage of frame-based imaging is its shorter data acquisition time. As a result, the latter can capture continuous hyperspectral cubes. Due to its non-scanning imaging, frame-based imaging has a more rigid image geometry, and it is possible to avoid the artefacts caused by movement. Moreover, because of the hyperspectral cube sequences are captured during flight, it is possible to estimate 3-D scenes based on SfM technique [106]. These types of sensors, indeed, combines the high spectral resolutions of line-scan hyperspectral imagers with a rapid collection of data [107].

More in general, the main limitations of light-weight hyperspectral cameras are related to the spatial resolution and the spectral range. Indeed, the spatial resolution is reduced to reach the high spectral resolution as specified above, and it is lower than the RGB cameras. The spectral range is constrained from 400–1100 nm or 1100–2500 nm due to payload limit compared with airborne cameras. As a new generation technology, indeed, hyperspectral cameras in RS are less compatible with drones compared with the RGB and multispectral sensors. Indeed, there are only a few commercial UAV-compatible that can mount these sensors (e.g. DJI matrice 600) most of all due to weight limitations; however also, in this case, the gimbal has to be customized ad hoc.

Table 6. Comparison of UAV's hyperspectral sensors.

Products	Resonon	Bay-spec OCI-F-HR	Headwall Nano-Hyperspec	Senop-Rikola	Senop HSC-2	Specim AFX10
System acquisition	push-broom	push-broom	push-broom	frame-based	frame-based	frame-based
Spectral range	400- 1000 nm	400-1000 Nm	400-1000 nm	400-1000 nm	400-1000 nm	400-1000 nm
Bandwidth	2.1 nm	3 nm	6 nm	10 nm	10 nm	5.5 nm
Pixel size	5.86 μm	[-]	[-]	5.5 μm	5.5 μm	[-]
Image size	900 pixel	800 Pixel	640 pixel	1010×1010 pixel	1024×1024 pixel	1024×1024 pixel
N of spectral bands	281	240	270	100-380	Up to 1000	224
Focal length	[-]	[-]	[-]	8.9 mm	[-]	15 mm

Moreover, due to their inner configuration, geometric calibration and band co-registration are other critical factors in the use of these sensors. For example, some studies are developed for the geometric calibration on the hyperspectral camera. However, the method is not standardized, and the repeatability in different environmental conditions is not guaranteed. De Oliveira et al. in [108] proposes the calibration of the Senop hyperspectral camera; however, few bands are calibrated, and the results showed that the internal orientation parameters are different for each channel. Honkavaara et al. in [109] describe a case study for band co-registration of the Senop hyperspectral; nonetheless, the results are not definitive. Moreover, it is worth to underline that the producer of hyperspectral camera does not provide software packages that allow the geometric calibration and the band co-registration. Some Matlab® or R routine or C ++ software are designed ad hoc, but they are not open source [109]. These are aspects that require further studies, thus, this thesis deals with geometric calibration and band co-registration of an hyperspectral camera in 3.2.3 and 3.2.4. Table 6 synthesizes the main commercial hyperspectral cameras, with a spectral range between 400-1000 nm used for UAS and UGS [88].

2.4 Sensor characterization

As explained above, after the platform and sensor definition, it is important to proceed with the sensor characterization. The process of characterization involves three attributes of sensor: spectral, radiometric and spatial. Spectral characterization allows a quantitative assignment to spectral sampling as described by the instrumental profile. The radiometric characterization enables an absolute scale to be applied to data amplitude. The spatial characterization describes the geometric relation of the imaging process of the camera, through the computation of the interior orientation and lens distortion parameters [110], [111]. Thus, the characterization process passes through the following main steps: spectral calibration (section 2.4.1), dark subtraction and vignetting correction (section 2.4.2), radiometric calibration (section 2.4.3), geometric calibration (section 2.4.4), and band-to-band registration (section 2.4.5) [112].

2.4.1 Spectral calibration

Spectral (or wavelength) calibration is a process for defining the spectral response function (SRF) of all spectral channels, and then control the spectrum stability and calibrate the wavelength shift based on the standard spectrum signal [57], [113], [114]. In detail the SRF assume a Gaussian shape and it is described with the center of wavelength and full width at half maximum (FWHM)[3] (Figure 9). The FWHM identifies the channel bandwidth, reported in microns or nanometers [116], [117].

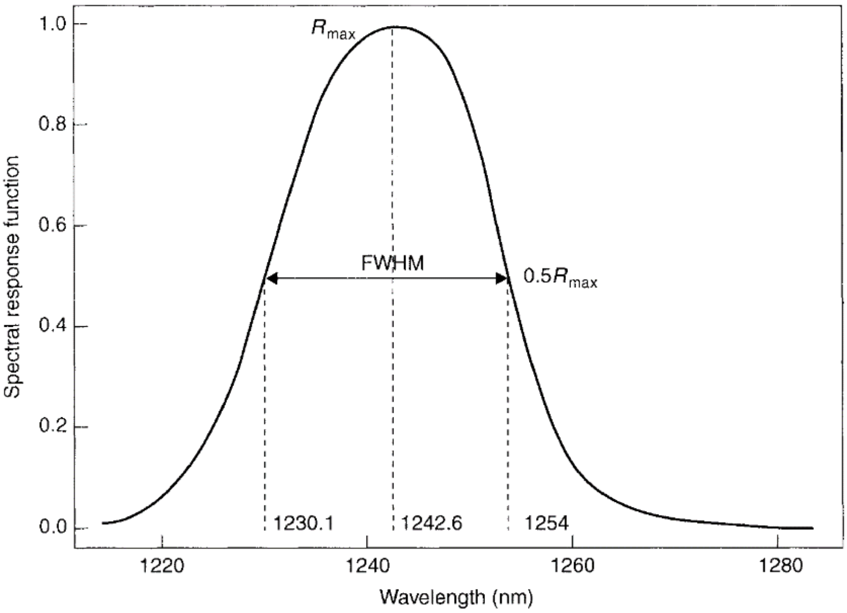


Figure 9. Spectral response function: the central wavelength (R_{max}) and FWHM [84].

[3] In a distribution, full width at half maximum is the width of a spectrum curve measured between those points on the y-axis which are half the maximum amplitude [115].

The instrument for the spectral calibration in particular for dispersive sensors has these components: (1) a light source combined with a monochromator that separates the wavelengths; (2) a collimator that changes the monochromatic light into parallel light producing a beam that fully illuminates the entrance pupil of the imaging spectrometer, and (3) a monitor that measure the variation of radiance from the spectral calibration source (Figure 10).

Light sources examples are monochromatic laser sources or pencil-style calibration lamps that produce narrow and intense peaks at a few known wavelengths. Argon, krypton, neon, xenon, mercury-neon, and mercury-argon are the popular gas types of commercial calibration lamps. A linear or non-linear regression analysis is used to predict the wavelengths at the unknown spectral channels. The monochromator has to be also spectrally characterized, usually by employing spectral line sources, often called pen-ray lamps. These are low-pressure, cold cathode discharge lamps with either a single gas, such as xenon, or a mixture such as mercury and argon. It is worth to notice that some instruments replace the collimator with Lambertian reflector.

The calibration procedure consists of scanning the monochromator to various wavelengths with an imaging spectrometer measurement performed at each monochromator setting. Before the inspection of the instrumental profile amplitude, the spectral characterization data are pre-processed performing a dark subtraction.

In general, in particular for UAVs multi and hyperspectral cameras, the manufactures perform, in the laboratory, this procedure before releasing the sensor to the users. However, an ongoing assessment could be useful to ensure the system's stability.

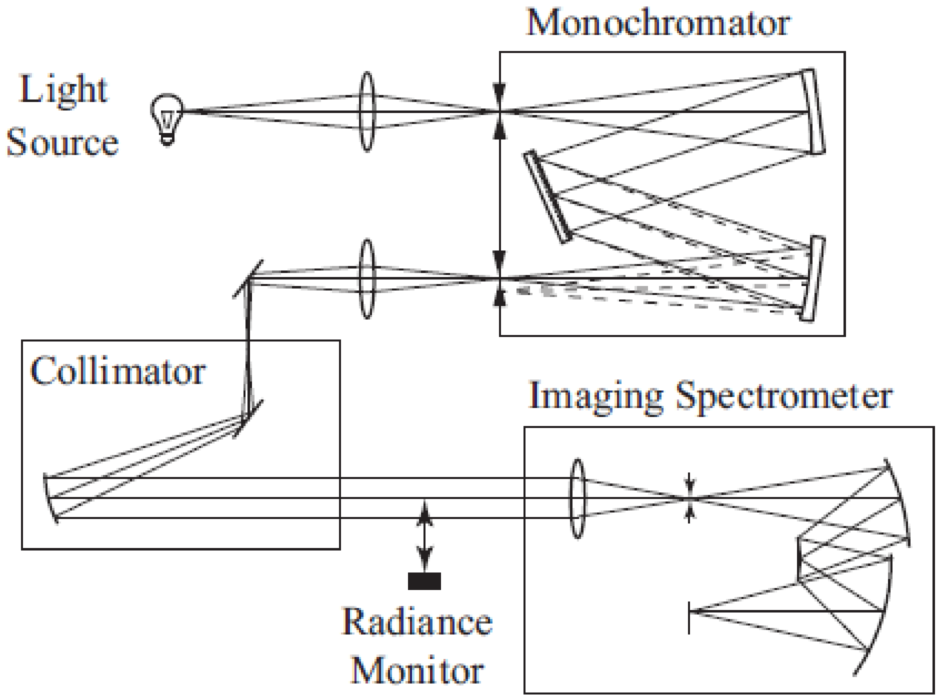


Figure 10. The spectral characterization apparatus [57].

2.4.2 Dark current correction and vignetting correction

The Dark current and vignetting corrections allow to transform the DNs in normalized (DNs)N, which the homogeneous response over the entire image during the operational time [118].

The dark subtraction mainly consists of taking away noise and thermal noise, which are related to sensor temperature and integration time. The procedure corrects the DN, estimating the dark signal non-uniformity (DSNU). The DSNU compensation can be performed in two different ways: by the thermal characterization of the DSNU in the laboratory at multiple integration times, or by dark current measurements during operation utilizing so-called "black pixels" within the sensor, or black image that are taken before each data acquisition closing shutter images.

The vignetting refers to a radial brightness attenuation of an image due to the effective lens aperture [119]. To normalize these effects, modelling the optical path or image-based techniques can be performed.

These two procedures can be considered part of the radiometric process because they contribute to the radiometric sensor characterization.

2.4.3 Radiometric calibration and correction

The radiometric response of an object is affected by different aspects, such as:

- the measurement geometry: the relative position of the object, the sun and the sensor;
- the illumination conditions: the illumination can be direct or diffuse;
- absorption and scattering of the atmosphere on the path from the object to the sensor;
- the sensing systems itself: vignetting and response function of the optical path and the chip.

The estimation and, when it is possible, the normalization of these effects is a requirement to define the reliable signature of an object. The radiometric calibration not only has the core function to characterize the sensors as an additional step after the spectral calibration but also allow to eliminate all the aberration of the radiometric signal related to the environmental conditions. The spectral calibration as explained in the section 2.4.2 aims at identifying the spectral response of each band, instead of the radiometric calibration refers to the definition of the relation among the digital number, the raw data collected by the sensor, and the physical units of radiance [W m^{-2} sr^{-1} nm^{-1}], and the subsequent conversion of radiance values to reflectance at the surface. The base principle is the evaluation of the DN response of the systems using reference with well-known radiance at various intensity levels to estimate the radiometric calibration parameters and the reflectance of the object.

Therefore, it is possible to identify few further steps after the spectral calibration: (1) sensor radiometric characterization, (2) reflectance factor generation, (3) radiometric scene normalization, (4) radiometric validation. Thus, first, sensor radiometric characterization allows transforming the digital numbers of the sensor to normalized $(DNs)_N$ as described in the section 2.4.2, and a further passage can be carried out to generate at-sensor radiance (Ls). Then, the data are converted in reflectance factors (R) with the empirical line method (ELM) using a radiometrically calibrated support device on the ground or in flight, at this point the scene can be reconstructed, or the image can be used as-is, and the reflectance factors can be converted reflectance quantities. At this stage, bidirectional reflectance distribution function (BRDF), shadows and topography effects could be corrected. The results of the radiometric can be validate using Lambertian targets that have not been previously used for the ELM. The step (2) is useless when the reflectance is not required. Moreover, step (2) and (3) can be swapped or carried out simultaneously [7] .

As in the case of geometric calibration, it is possible to identify different strategies for full radiometric calibration: laboratory, on-board, test field (vicarious) and self-calibration (on-the-job). The main difference among the methods consists of conditions in which calibration takes place. This influences, as consequences, the equipment to use and the level of accuracy.

- *Laboratory calibration*: is performed indoor using typically integrating spheres or hemispheres as light sources [120]. This procedure can be carried out periodically, simultaneously with spectral calibration. In details, the sensor is exposed to a homogeneously illuminated target to normalize for the non-homogeneous illumination of the chip due to the optical path of the system (e.g., vignetting) and differences in the radiometric response function of the individual elements of the chip. The result is a correction function that transforms the DNs recorded by the chip to linear radiometric coefficients. Furthermore, it is possible to identify the response of the sensor in radiance when the light source is known.
- *On-board calibration*: is carried out in fight conditions utilizing various onboard calibrators or natural light sources (the Sun, the Moon).
- *Vicarious methods*: is achieved in flight determine the system calibration in flight conditions using artificial targets present in the scene, or natural targets, such as desert sand or salt flats. An alternative to the reference target is the use of incident irradiance. The vicarious calibration can be the radiance-based or the reflectance-based method.

In the first case, a well-calibrated radiometer measures the radiance of the ground target [121], [122]. In the case of the reflectance-based method, accurate information on atmospheric conditions and object reflectance are available.

- *Self-calibration also called radiometric block adjustment:* as, in the case of geometric calibration, is performed using the actual mapping data. The procedure uses some optimization techniques exploiting the redundant information from multiple overlapping images to model the function between the DN and the object reflectance. In the process, tie points are identified in various images and radiometric control points (i.e. reference panel) can be included. The outputs are the parameters of the radiometric model that can be applied to produce radiometrically corrected images[123]. This procedure is implemented in the most common SfM software.

Among the radiometric method, the laboratory calibration is the most rigorous due to the stationary and controlled conditions in which the operations are performed. The use of UAV, comparing with airborne or satellite platforms, has as the main advantage in radiometric measurements the possibility to fly below the clouds. However, the movement of the cloud can cause problems [124]. Thus, unstable atmospheric conditions and the stability of irradiance can influence the calibration in-flight. For low-altitude remote sensing the vicarious calibration, in particular reflectance-based methods, and self-calibration approach are the most used. According to Aasen et al. [118], the self-calibration is the technique that better perform the calibration in unstable atmospheric and irradiance conditions. Instead, the accuracy of vicarious methods resent of unstable irradiance [118]. Is worth to notice that, mathematical models that describe the radiometric process using the vicarious procedures are available to users; the self-calibration instead can be performed using *ad hoc* software that works as a "black box" (e.g., Pix4D MapperPro), if it is necessary [125].

As the base of these strategies [123], in general, to generate reflectance images from the radiance images, two approaches can be defined:

- irradiance measurements, based on a second sensor (or optical path) that measures the spectrally resolved downwelling illumination [126];
- radiometric reference targets. Targets should have a uniform intensity and be close to Lambertian reflectance characteristics. The calibration targets should be flat and levelled, without obstructions, and should be large enough (preferably more than five times the image GSD) to reduce adjacency effects by only selecting the middle part of the panel [124].

The incident irradiance can be either estimated using atmospheric radiative transfer models (ARTMs) or measured using an irradiance spectrometer. ARTMs allow simulating the incoming irradiance from the sun to the top of the object. At the same time, they enable simulating the atmosphere influence on the signal during its path from surfaces to the sensor. Input parameters include the time, date, location, temperature, humidity, and aerosol optical depth measured by, e.g., a sunphotometer. The drawback of ARTMs for reflectance calculations is the need for sufficient parameterization of the atmosphere. This is particularly challenging for flights over larger areas, where the atmosphere might be heterogeneous, and under varying illumination conditions due to clouds.

As an alternative to the ARTMs, it is possible to use a stationary or mobile irradiance sensor with cosine receptor optics. The first sensor can collect consecutive measurement of the Lambertian reference panel on the ground; the second can continuously measure the irradiance in-flight or during the ground operations [127], [128]. When irradiance measurements based on a second sensor are considered, these measurements can be employed to transform radiance measurements of the target objects to reflectance by dividing radiance by irradiance after both sensing systems have been cross calibrated for a spectral and radiometric response. It is possible to make two assumptions: the atmosphere between the ground and the sensing system does not influence the signal; the illumination is the same for both. A useful check can be comparing signals over the calibration panels at the ground and the flight levels at (or near) real-time. Ideally, the irradiance sensor is also mounted on the platform (UAV or UGV) at a reciprocal angle to the measurement geometry of the sensor (if a sensor is pointing nadir, the irradiance sensor should be pointing zenith).

When the irradiance measurement is not available, several (near-) Lambertian targets with a known spectral reflectance and ELM is commonly used to calculate reflectance factors (HDRFs). Thus, the procedure allows to generate reflectance and normalize signal for different illumination conditions between flights and for atmospheric effects (although the atmosphere between the ground and UAS may only have limited impact on low altitude for standard reflectance measurements).

Software packages such as Exelis Visual Information Solutions (ENVI) developed by L3HARRIS implements the method in its tool [129]. After the conversion from DN to radiance, the technique combines the linear conversions from the DNs to the reflectance factor into a single linear transformation. ELM is the most common and straightforward approach. It is suitable for survey times shorter than 30 min under stable weather conditions (clear sky) when ELM results of panel measurements at the beginning and end of a flight were linearly interpolated (Miura and Huete [158]). The covering of the reflectance range of interest has to ensure with a minimum of two reference targets. This is possible when the distribution of DN to reflectance within an image can be considered linear [130]. Several attempts to use a simplified ELM, reducing the number of the targets to one (generally the white target) or using pseudo-targets [130] are made in the literature [131]–[133]. It shows some problems when UAV flies above the reference target at a short distance [134]. Adding more than two targets reduces uncertainties, enables an assessment of sensor linearity, and allows evaluation of the ELM results by using some panels only for verification.

Moreover, one-point empirical line calibration, where the sensor collects a single point above of one reference before or after take-off, should be avoided. Therefore, this produces biases related to the shading of the hemisphere.

As cited before, to conclude the procedure of radiometric calibration, also the radiometric scene normalization has to be performed, in particular, BRDF, topography, and shadows have to be corrected.

The BRDF correction refers as the process of compensating the influence of anisotropy so that the image reflectance values correspond to the reflectance factor at the (mostly) nadir direction. Indeed, in the analysis of data captured by the wide-angle field of views sensors, the anisotropy propriety of the materials might cause significant radiometric differences within individual images and between neighboring images. This affects, indeed, the spectral signature of objects within the scene and produces unwanted effects during the process of image mosaicking. The standard classification of BRDF-models divided them into physical, empirical, or semi-empirical approaches. However, in low altitude remote sensing applications, simple empirical models are usually used for a BRDF correction. In the studies of Walthall et al. [135] and Nilson and Kusk [136] and Honkavaara et al. [123], [137] it possible to find some applications.

The topography corrections are related to slope orientation and exposition effects due to the position of the objects respect to the sun. For the correction, Digital Surface Model (DSM) and the sun's elevation and azimuth angles at the time of acquisition are the necessary input data. Shepherd and Dymond report the main standard methods in satellite application [138]. Some examples are Lambertian methods, such as the cosine method, gamma method, as well as non-Lambertian methods, such as the Minnaert method or the c-factor method [118]. However, Jakob et al. [139] implemented and tested these approaches in the UAV-applications.

Finally, the shadows are caused by 3D objects within the scene and created by clouds. In satellite and airborne applications, de-shadowing approaches can be classified into histogram thresholding, invariant colour models, object segmentation, geometrical methods, physics-based methods, and unsupervised and supervised machine learning methods [140]. As reported by Aasen et al. [118], however, no applications of the de-shadowing exist in for UAV-imagery processing.

2.4.4 Geometric calibration

Geometric (spatial) calibration is a process that correlates points in world coordinates and their corresponding image locations [141]. This means knowing the internal geometry of the camera, and, thus, the estimation of Internal Orientation Parameters (IOPs) is crucial. The results of the calibration consist in the assessment of the principal point PP (*x_0, y_0*) in the image coordinate system, focal length (*c),* the polynomial coefficients or radial distortion and tangential distortions (*k_1, k_2, k_3, P_1, P_2*) and the skew [142].

The following mathematical model describes radial (eq. 1) and tangential distortion (eq. 2, 3) [119]:

$$\Delta_\rho = k_1\rho^3 + k_2\rho^5 + k_3\rho^7 \ldots \tag{1}$$

where

$\rho = \sqrt{(x - x_0)^2 + (y - y_0)^2}$ is the radial distance;

x,y: image coordinates;

k_1, k_2, k_3: coefficients of the radial distortion. It is possible to notice that k_1 is multiplied by the cubic radius; thus, it generally can be considered the most significant. *k_2, k_3*, instead, are representative of wide-angle lenses or accuracy enhancement.

$$\Delta t_x = P_1 \left(\rho^2 + 2(x - x_0)^2\right) + 2P_2 (x - x_0)(y - y_0) \tag{2}$$

$$\Delta t_y = P_2 \left(\rho^2 + 2(y - y_0)^2\right) + 2P_1 (x - x_0)(y - y_0) \tag{3}$$

where

$\rho = \sqrt{(x - x_0)^2 + (y - y_0)^2}$ is the radial distance;

x,y: image coordinates;

P_1, P_2: coefficients of the tangential distortions.

According to the literature[111], [143], camera calibration can be achieved using different strategies:

- The *test-field calibration* is performed using a gridded panel (i.e., with black and white chessboard patterns) with well-known coordinates or distances (Figure 11). The procedure follows the close-range photogrammetry rules: (1) images of the panel have to be collected from different positions, (2) the chessboard has to cover the entire sensor format and (3) an adequate transversal, and longitudinal overlapping between sequential images has to be guaranteed [143], [144]. In general, it is suggested to use 10-20 images with different distances and orientations. An appropriate acquisition geometry is important to avoid undesirable correlations between parameters. After the image collection, semi-automatic or automatic procedures are used for detecting angles of the pattern.

This method is used when the camera has to be calibrated separately from three-dimensional object reconstruction. In this case, the mechanical stability of the camera itself determines the validity of the calculated parameters directly.

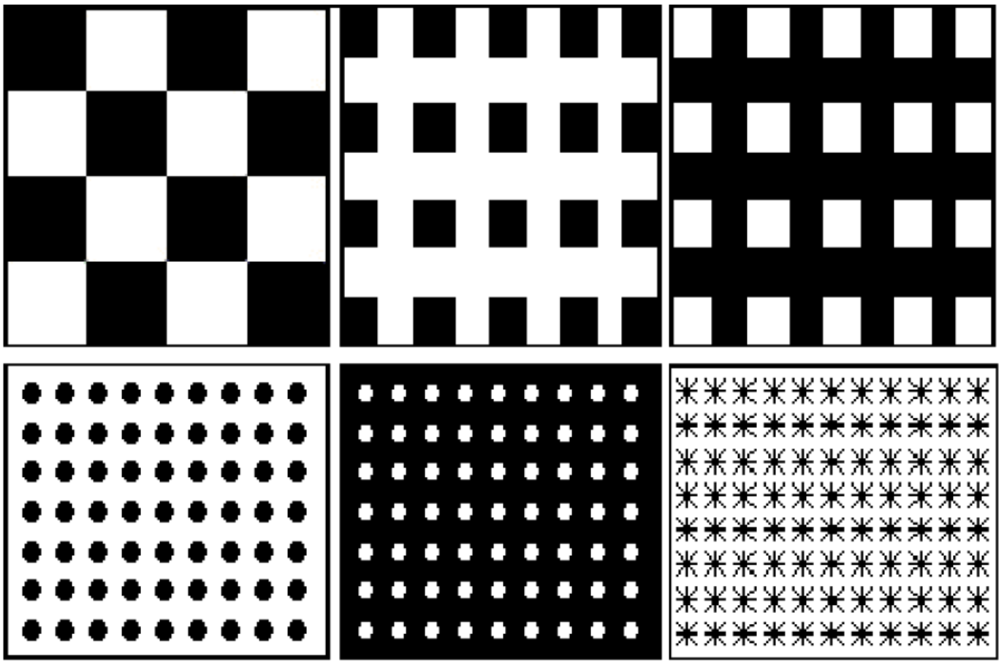


Figure 11. Example of patterns for gridded panels [145].

- *On-the-job* calibration consists of determining the parameters of the internal orientation and additional parameters using a series of images of the survey's object, and the parameter estimation takes place simultaneously with the object's three-dimensional reconstruction. Appropriate three-dimensional reticular structures are often used on which control points are placed.
- *Self-calibration* simultaneously returns both calibration parameters and reconstruct the model using the images. The self-calibration can be performed with three different procedures:

(1) in a laboratory with a high precision 3D calibration field, with well-known Ground Control Points (GCPs). In general, 3D calibration fields can be built on a building façade, as reported in [146] and [147]. As in the case of *test-field calibration,* the acquisition during the calibration tests has to consider convergent

images taken from varying 180° viewing directions and different heights and orientations;

(2) on-the-field using the directly the images with more GCPs. In this case, the object itself replaces the calibration panel. The convenience to use this approach is that the internal orientation parameters are determined simultaneously with the object's survey. All the observations on images, collected with different perspectives, and the redundancy of the GCPs contribute to the estimation of the unknown parameters;

(3) based on CV algorithms and feature extraction using tie-points. It is worth noting that the focal length cannot be estimated in this case.

To perform the calibration, CV-based software tools can be used. Most of them use predisposed planar patterns with well-known individuated points (chessboard is the most frequent configuration). An example of calibration through chessboard patterns can be found in the Matlab® "Camera Calibrator" toolbox [148]. The toolbox contains algorithms for the pinhole camera model [142] and fisheye camera model. It allows easy estimation of the camera parameters through an automatic procedure that requires the chessboard square dimension and the images of the chessboard pattern taken by the camera as inputs. The toolbox default settings include intrinsic parameters and two radial distortions; however, it is possible to calculate the radial distortion third coefficient, the tangential distortion coefficients and the skew value.

An alternative solution nowadays is provided by SfM software, such as Agisoft Metashape [149] or Pix4D [150], for the 3D point clouds production by images. Therefore, this software can automatically extract these parameters after the image alignment. It is worth underlining that Structure from motion software is not developed enough to manage hyperspectral data. Indeed, even if they can process multispectral images, they cannot sustain the HSI huge dimensionality, and the IOPs estimation and three-dimensional reconstruction can be processed per band.

2.4.5 Band co-registration

Due to the acquisition system configuration of some spectral sensors, the band-co-registration is necessary. The band co-registration is the process to align bands of a single data cube that do not overlap perfectly. Junior et al. [151] underline the main issues to carry out this process. First, each channel's spectral content is not the same because objects respond differently in each part of the spectrum. Moreover, some spectral sensors use multiple sensors to obtain spectra and the physical displacement causing misalignment among channels.

The approaches for band-to-band registration include:

(1) the use of Real-Time Kinetic (RTK) GNSS/GPS with Inertial Measurement Unit for a direct orientation;

(2) the georeferencing of the band separately using the bundle-block adjustment. Image blocks with typically 20-48 bands have to be processed [109]. This method requires a long processing-time when more than 100 bands have to be included in the block adjustment calculation;

(3) the employment of a computer vision approach to calculate the transformation parameter from reference (anchor) bands. In this case, a single image can be used to evaluate the transformation [152]. In particular, the CV-based method consists of establishing correspondences between different bands through local feature descriptors.

In [109], [153]–[155], it is possible to find some practical applications of band co-registration for multispectral and hyperspectral sensors.

Since approach (1) is an expensive solution [156] and (2) is a very time-consuming procedure, the use of the CV approach can be the most effective solution. The CV- based band co-registration workflow consists of three steps as follows (Figure 12):

(1) Definition of anchor bands: the band co-registration function is applied to a band-pair (reference-target images) at a time. According to the literature [109], the number of reference bands should be 1-3 typically. The bands, in this way, can be divided into two or three groups with different anchor bands. The selection of reference bands could consider two different band-wise arrangements: the spectral order of ascending wavelengths or the temporal order in which the sensor acquires the band. For instance, in the spectral order for VNIR sensors, the reference bands could be chosen according to two spectral groups: the green-yellow (550–580 nm) and red-edge (660–700 nm) regions. In the case of temporal order, the reference bands could be taken in the beginning, middle and end of the data cube. In any situation, the first band is used as the anchor, and the next bands are used as the target.

(2) Estimation and extraction of key points for each pair of bands: an image-matching algorithm can be used to extract key points in anchor-targets and calculate their position, orientation and scale. Binary Robust Invariant Scalable Key points (BRISK) [157], Maximally Stable Extremal Regions (MSER) [158], Harris-Stephens Features (HSF) [159], Scale Invariant Feature Transformation (SIFT) [160], Speeded-

Up Robust Features (SURF) [161], and KAZE Features [162] are some implemented examples.

(3) Estimation of transformation parameters: translation, scale, and rotation are computed using the matching key points calculated in step (2). Thus, the target image is transformed into the anchor reference system using an affine transformation model in most cases [109], [156].

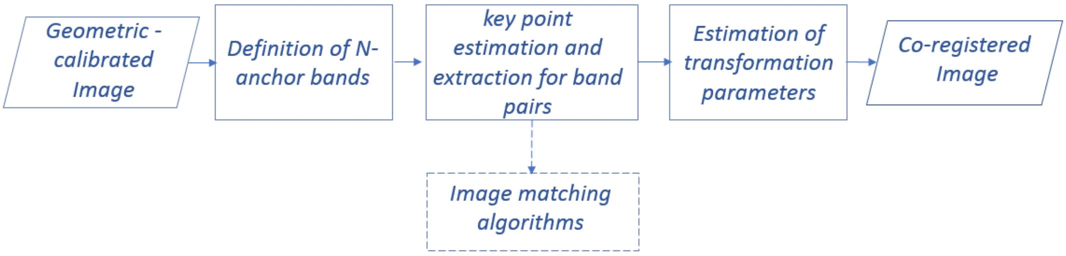

Geometric - calibrated Image
Definition of N-anchor bands
key point estimation and extraction for band pairs
Estimation of transformation parameters
Co-registered Image
Image matching algorithms

Figure 12. CV-based band co-registration workflow.

Chapter 3

System configuration and calibration

The literature review in Chapter 2 outlines that platforms and their sensors must be chosen based on applications and their level of detail. The information accessible from RS platforms depends on the features of the platform/sensors system. Therefore, it is fundamental to analyze all the UAS components and investigate the application-specific configuration of the platform. Moreover, in monitoring and inspection task, the payload configuration that comprises the imaging sensors plays an important role. The selection, and the characterization of the imaging sensors is, thus, an essential step.

As mentioned in Introduction (Chapter 1), this thesis investigates the development of IE tools for real-time monitoring and inspection, starting from the platform definition to the information extraction algorithms. This chapter focuses on the configuration of an aerial platform for the two identified applications. A multi-rotor UAV was chosen in the thesis because it allows integrating payloads heavier that the fix wings and it is easier to customize. Two applications mean different platform/sensors system for data acquisition. In details, for the aircraft de-icing system, the use of spectral sensors is required.

Particularly challenging was the definition of the UAV, equipped with the light-weight hyperspectral sensor for real-time operations. Indeed, as explained in Chapter 2, RGB and multispectral sensors already embedded on UAVs, while the hyperspectral camera needs further effort for their integration. Even if the hyperspectral sensors for UAV application are an emerging technology, their integration on the UAV is fully not developed yet. On the market, only a few platforms, commercialized only the last year, proposed embedded solution for this type of sensor. The problems related to the integration includes the gimbal, i.e., the device for keeping the hyperspectral camera in-built with the drones, and the system for remote control of the camera and access to the data. Some models of the hyperspectral cameras allow the acquisition in a memory card; however, to control and processing data for a real-time solution, the data have to be accessible. Thus, a way to download data in a remote ground station has to be investigated.

The issues related to the hyperspectral camera concern also the sensor characterization and calibration. In general, the geometric and radiometric features of sensors can be provided by the producer or a proprietary software associated to the sensors can be released by the manufacture and sold with the sensor to manage these aspects. However, for the specific hyperspectral sensor chosen, the characterization of the sensor needs further analysis.

Therefore, first, this chapter analyses the UAV components of the systems employed in this research (section 3.1). Then, an in-depth study of the selected hyperspectral sensor is presented from a geometric and radiometric point of view, pointing out the main gaps reviewed in the literature (3.2).

3.1 System configuration

A UAS is a complex system composed of the three main components that work coordinately [163]. The aerial platform, which includes the airframe, the navigation system, the power supplies and the payload; the ground control station (GCS) that enables the remote human control; the communication system which allows the link between the UAV and the GCS (Figure 13).

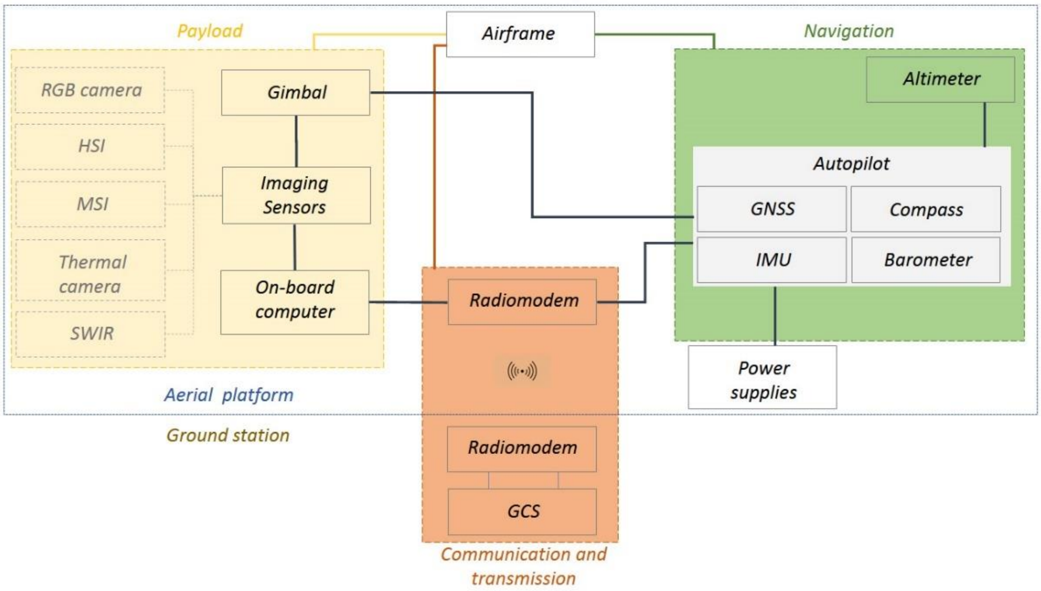


Figure 13. UAV main components.

The aerial platform includes four main subsystems whose purpose is to allow the flight and carry the sensors for data acquisition:

(1) the airframe: the main structure of the UAV. This structure has to be light-weight, aerodynamically efficient, stable and compliant with the sensor's weight. Fixed wings on UAVs are mainly made on polystyrene

or plastic, while standard multi-rotors airframes are in aluminum or carbon fiber.

(2) the navigation system: the main module of this subsystem is the autopilot. The autopilot is a small and light hardware/software component that enables to perform an automatic flight by the input of different sensors. The hardware component is a circuit board with various connecting ports for sensors. The set of sensors encompasses positioning sensors such as the GNSS and the attitude sensors, i.e. IMU, barometer and compass.

(3) the power supplies: the module supplies energy to the system. Based on the airframe, different types of power supply system can be adopted, e.g. Wankel rotary engines, fuel cells or electric UAVs [164]. In details, the power supplies of multi-rotors systems are batteries, such as Lithium Polymer (LiPo) batteries. The main features of LiPo batteries are voltage, capacity, discharge rate and downside.

(4) the gimbal: the device stabilizes the camera on the UAV. It allows pointing the camera in different directions. Gimbals are mounted on every drone used for photogrammetry and remote sensing applications. The gimbals are controlled by either an operator or an on-board computer. Gimbals are mounted underneath the vehicle's body, and the most sophisticated contains an IMU which is mounted on the same plane as the camera, to measure the attitude [165].

(5) the payload: the set of sensors to gather the information that can be partially processed on-board or transmitted to a base station. The payload includes the imaging module that can include, the RGB cameras or spectral cameras.

(6) the computing module: this module includes the on-board computer system. It can control the payload's sensors, and process and storage data. This component is handy for real-time procedures because it works on planning and decision making. As an alternative to the computing module, only the storage module, i.e. the memory card can be used for saving data on-bord. In this case, the data are post-processed after the end of the flight.

The ground control station is a computer or a tablet on-the-ground for monitoring the mission and interact with the payload sensors. This subsystem is controlled by the pilot and allow to manage the flight parameters (e.g., flight hight, coverage) by a mission planning software.

In the end, the communication and transmission module guarantees the continuous link between the UAV and the ground station and allow transmission of information. This module is essential because it allows both the control of the navigation module and the access to the data collected by the payload. The data are transmitted from UAV to GCS in a bidirectional way by a datalink, that can be uplink or the downlink. The uplink is the link from the GCS to autopilot. The downlink is the telemetry link from the autopilot to the GCS. Two radio modems, the on-board and the on-ground segments form a standard wireless data transmission module. These both can work on the same frequency and usually have the capacity both to transmit and receive data. For decades, Pulse Width Modulation (PWM) in the Very High Frequency (VHF, between 30 and 300 MHz) band has been used. Interferences can easily affect this type of communication.

The Internet Protocol (IP), based on Wide Local Area Network (WLAN) digital connection or, the Long Term Evolution (LTE), 4th Generation (4G) channel are the next-generation solutions. Both of these ways of communication are interesting solutions for their effectiveness and their costs [166].

3.1.1 UAV platforms and system components

For this research, ready-to-fly (RTF) platforms were selected. The use of a ready-to-fly platform allows focusing only on imaging and the computing modules for real-time applications. Moreover, the full design of the drone is out of this dissertation. The DJI products were chosen because targets researcher and UAV-developer, they are compatible with a large amount of the sensors on the market, and they are customizable. In details, two multi-rotors UAS were selected: DJI Phantom 4 Pro and the DJI Matrice 210 v2. The DJI Phantom 4 Pro was employed for the RGB and multispectral acquisition, while the DJI Matrice 210 v2 was used for the hyperspectral camera. Table 7 summarizes the specification of the selected sensors. According to the classification proposed in Chapter 2, the DJI Phantom 4 Pro is a small size drone, instead of the DJI Matrice 210 v2, that is a large size UAS. It is clear, thus, why two platforms were used. The payload of the DJI Phantom 4 Pro does not support the hyperspectral sensor.

Table 7. Imaging module: RGB and spectral sensor specifications.

Camera	RGB FC330	RGB ZenMuse XT2	MAPIR Survey3N	Senop Rikola
Lens optics	FOV 94° 20 mm	FOV 57.12°x 42.44°	41° HFOV (47mm) f/3.0 Aperture	H 36.5°, V 36.5°
Spectral range (nm)	-	-	550-850	500 – 900
Spectral resolution (nm)	-	-	40	10
Spectral channels	3	3	3	380
Shutter type	Global	Global	Rolling	Global
Focal length (mm)	3.61	8	8.25	9
Image resolution (pixel)	4000 × 3000	4000 × 3000	4000 × 3000	1010 × 1010
Pixel size (μm)	1.56	-	1.55	5.5
Mass (g)	-	-	76 g	720 g
Cost	-	-	~700€	~60,000€

The DJI Phantom 4 Pro is a quadcopter (Table 8). It includes positioning and attitude sensors, a radio transmitter and the autopilot board. The imaging module includes an RGB sensor, i.e. RGB FC330 (specifications in Table 7). The RGB sensor is completely integrated. It is managed by the autopilot software that triggers images according to the flight plan, turns-off the camera during the take-off and the landing. However, this UAV allows adding on the payload other portable sensors such as MAPIR Survey 3N (Table 7), a multispectral camera, thanks to an easy-to-use mount kit provided by MAPIR. The mount kit allows to attach the additional camera without removing the original RGB form the drones [167]. It is not possible to mount the MAPIR on the DJI Matrice 210 v2, because the kit is not available.

Table 8. DJI Phantom 4 Pro specification.

Technical Specifications	
Weight	ca. 1.39 kg (with batteries and wings)
Maximum payload	< 1 kg
Flight time	30 min (with maximum weight)
Maximum altitude	6000 m
Data transmission range	7 km
GPS/GNSS	available
Imaging module	RGB and multispectral camera compatibility
Imaging processing	On-board computer not available

The DJI Matrice 210 v2 is also a quadcopter. Table 9 summarizes the specifications [168]. This platform was selected because it is customizable and additional equipment can be integrated with it. Moreover, an on-board computer completely embedded with this platform has been announced by DJI, Manifold 2, but not yet distributed in Italy [169]. Therefore, it allows further implementation. There is also the possibility to connect an Android system to the remote controller and run own-developed applications on these devices. For this purpose, the DJI released to developers a Software Development Kit (SDK) available for mobile applications.

Table 9. DJI Matrice 210 v2 specification.

Technical Specifications	
Weight	ca. 4.69 kg (with two TB55 batteries)
Maximum payload	1.34 Kg
Flight time	24 min (with maximum weight)
Maximum altitude	3000 m
Data transmission range	> 5 km
GPS/GNSS	available
Imaging module	RGB and multispectral camera compatibility. No compliance with a hyperspectral camera
Imaging processing	On-board computer not available

Since it is an RTF platform, some components such as the airframe, the autopilot, the communication module, the power supplies were already engineered by the producer. As mentioned above the payload is customizable; thus, the imaging module can be chosen according to the application. Two different sensors were used for this work on this platform, the DJI RGB ZenMuse XT2 and the hyperspectral camera, the Senop Rikola. Both cameras are described in Table 7. In details, DJI RGB ZenMuse XT2 is completely integrated with the drone. Instead, the hyperspectral camera required and in-depth study. The use of this camera, for real-time application, requires indeed, also the integration of the gimbal, the computing and storage module, the radiometric correction module, the transmission module and the additional power supply (Figure 14). The primary constraint that has to be respected in the configuration process is the maximum payload allowed, in this case, 1.34 kg.

For what concerns the Senop Rikola, further specifications on the camera and its characterization and calibration are presented in section 3.2. The Senop Rikola is equipped with GNSS receiver, for the image georeferencing process and the radiometric correction module, i.e. an irradiance sensor that measures downwelling irradiance, and it is useful for in-situ radiometric calibration. The camera operated in two configurations: connected to the computer (as a spectroradiometer) or in standalone mode with a memory card. The acquisition configuration can be chosen according to the approach considered to manage real-time procedure in monitoring and inspection. Indeed, two different methodologies can be applied, and the equipment depends on which one is selected. The first approach considers the possibility of onboard imaging processing and decision making. The second approach considers a two-step procedure: in-flight data acquisition and the image analysis on-the-ground. In the first approach, a light-weight computer must be included for real-time imaging processing and result visualization. Otherwise, in the second approach, a low delay and high capacity communication and transmission module has to be selected for downloading the data into the ground station. In this case, the first approach was preferred.

Therefore, according to the decision to proceed with in-flight image processing an additional computer must be included on the payload. For this purpose, an Intel Compute Stick 64 (64GB) of type STK2mv64CC was added to the system [170]. The weight of the computer is 60.2 gr. A component of the Intel Stick is the wi-fi module that allows the transmission of data to the GCS. The computer on-board can be alimented by batteries powering the motors of the UAV.

As mentioned before, there is not a ready-to-use gimbal for integrating the hyperspectral camera on the DJI Matrice 210 v2, and the customize gimbals are very expensive. Therefore, to use the camera on the drone, a gimbal *ad hoc* was designed at Politecnico di Torino. To produce a light-weight gimbal, the carbon fibre was selected as material [171].

Finally, for the safety of the sensor and the data, an additional battery was included in the system. A LiPo battery, 2S with a weight of 40.8 gr was chosen according to the camera specifications.

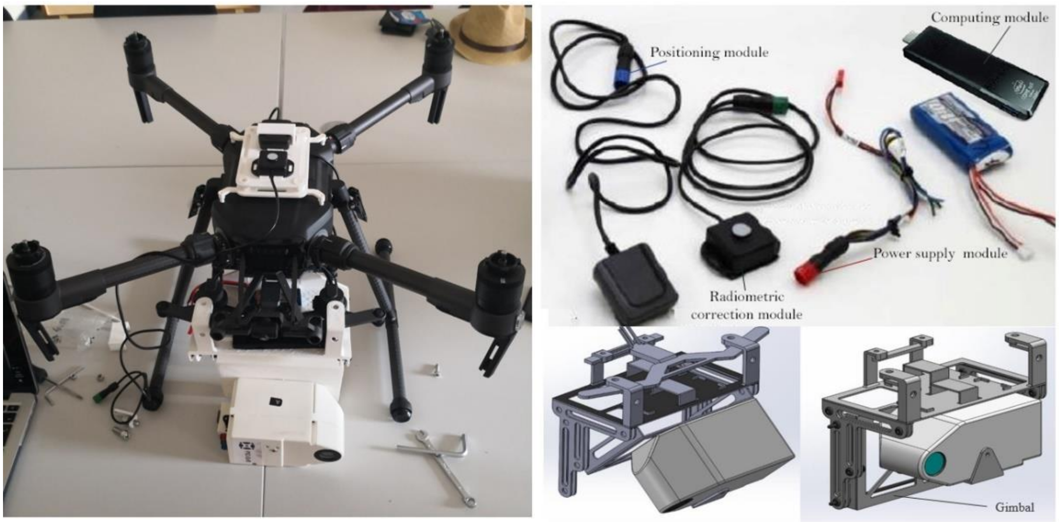


Figure 14. DJI Matrice 210 v2 configuration.

3.2 The Rikola camera characterization

As underlined above, despite the multispectral and RGB sensors, hyperspectral technologies are not yet fully embedded. As mentioned in section 3.1, among the sensors used for this research, the Senop Rikola camera is included [172]. Since the camera is employed in UAV application only starting from 2016, and the documentation related to the sensor characterization is still incomplete, some aspects related to the definition requires further analysis.

As explained in section 2.4, the sensor characterization passes through spectral calibration, dark current, and vignetting correction, radiometric calibration and geometric calibration and band co-registration. In section 3.2.1, a sensor description is provided and, starting from the gaps derived from the lack of documentation, and pointed out from the literature, a complete characterization of this particular sensor is presented in this work. Therefore, a detailed analysis of literature related to Senop Rikola sensor was carried out for each part of the characterization process to fulfil the necessity of complete knowledge of the sensor. It is worth to underline that the manufacture provides just the spectral calibration certificate and additional information related to the process of spectral calibration can be extracted from [173]. Moreover, the camera is equipped with basic software, HSI software, that helps to manage the dark current correction and vignetting. Thus, to use the hyperspectral data for UAV's applications, detailed information, and the procedure related to radiometric calibration, geometric calibration and band co-registration have to be retrieved.

For what concern the radiometric calibration, Honkavaara et al. [137] developed and tested a process for radiometric correction of UAV image block.

After the spectral calibration and the radiometric sensor characterization using calibrated lamps, the entire process is implemented in BAE Systems SOCET SET [174], [175], a photogrammetric environment with in-house developed components. Moreover, Honkavaara and Khoramshah [123] works on the optimization of the radiometric correction process assessing the feasibility of the method using multi-view images in agriculture application. The importance to use reflectance panels and knowing the irradiance measurements for radiometric block adjustment is underlined. Even if the way to use the reflectance panels is well explained, the way to introduce the irradiance measures in the process is not standardized. As stated by [124], the use of irradiance as input data for radiometric correction is challenging. Indeed, on flight measurement can affect the tilt of the sensor. Ad hoc gimbals can compensate for the errors; however, a ground spectrometer is required to achieve acceptable results. Hakala et al. [124] implemented a methodology for direct radiometric correction with irradiance, but they employed indeed also ground measurements. A test of direct measurement is presented in [176]. This literature review points out that self radiometric calibration is the most advanced method for correcting HSI block images and reflectance panel and empirical line methods remain the most used tools for radiometric calibration of hyperspectral images. The section 3.2.2 presents the radiometric inter-calibration procedure applied to verify the calibration of the camera used for this research and precise the tools used for the calibration of hyperspectral images in this work.

The geometric calibration of the Rikola camera was studied during the past year in different works. De Oliveira [108], proposed a self-geometric calibration procedure for this sensor. The work showed the process of calibration for only 24 random bands and pointed out slight variations of the IOPs for each band. Tommaselli in [50] presented an in-depth study of the variation of IOPs. To simplify the calibration procedure, Tommaselli et al. [177], applied a preliminary band-coregistration procedure for reducing the number of required IOPs.

Furthermore, the analysis proposed in [178], shows that on-the-job calibration produces relevant results in the calibration parameter estimation in UAV's applications, pointing out the necessity to study the stability of the IOPs and definition of reference IOPs. Therefore, analyzing these studies, no definitive IOPs for all the bands, or the two sensors of this camera are provided for correcting new data cube. Thus, further analysis to establish the IOPs' stability has been developed in this work and described in section 3.2.3.

Few papers report band co-registration of this type of camera. As reported in section 2.4.5, the main issue related to this step is related to the spectral variability among bands that complicates the feature extraction process. In general, groups of reference band with similar spectral features are used for matching the rest of bands to them [152], [179], [180]. More in-depth, in [179] SIFT and SURF algorithms are used for correcting misalignments using Matlab® routine combined with C/C++ code. Instead, in [180], in details, a preliminary study assesses the potential of various 2D transformations in-band registration in Fabry-Perot Interferometer (FPI) cameras and comparing the polynomial transformation, and the affine transformation identifies the first as a more suitable solution. In this study, the tool used to implement these test is not mentioned. Besides, Honkavaara et al. presented in [109] a rigorous procedure for 3D scene band coregistration developed in FGI' software using the C++ programming language. The FGI software is not open source as also the customized routine Matlab® and C/C++ code. Thus, the only way to co-registered the data cubes is following the procedure described in the literature and customizing the Matlab® routine to elaborate hyperspectral data. In the section 3.2.4 is described the Matlab® code customized for our applications.

3.2.1 Sensor description

As specified above, the SENOP Rikola camera is the hyperspectral sensor used in this work. The prototype camera, produced by Senop, is developed by the VTT Technical Research Centre of Finland [181], [182]. This frame-based camera is based on Fabry-Perot Interferometer, and different models are available for inspecting various part of the spectral range: 450-800 nm, 500-900 nm. For this research, the model that can measure the spectral range from 500 to 900 nm has been chosen (Figure 15). The fixed focal length of the camera is 9 mm, and the Field of View (FOV) is 36.5° in both directions. The camera can collect images using two different frame resolutions, 1010×1010 and 1010×648 pixels. The primary specifications are summarized in Table 7 (section 3.1).

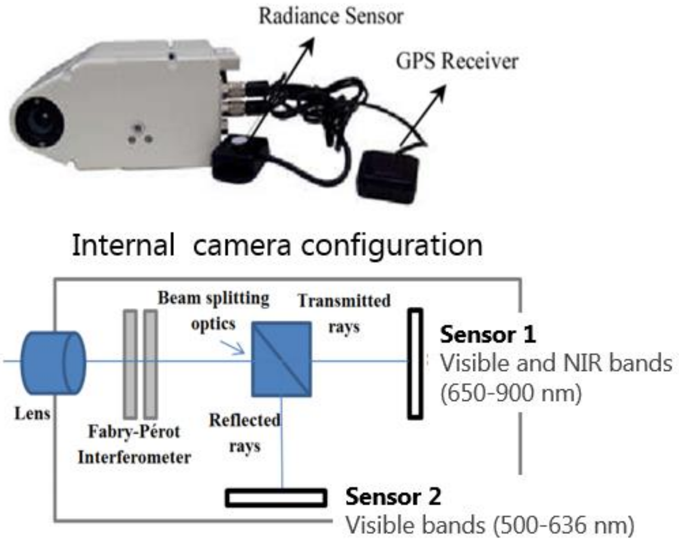

Radiance Sensor
GPS Receiver
Internal camera configuration
Beam splitting optics
Transmitted rays
Sensor 1
Visible and NIR bands (650-900 nm)
Lens
Fabry-Pérot Interferometer
Reflected rays
Sensor 2
Visible bands (500-636 nm)

Figure 15. The physical internal configuration of the Senop Rikola camera [108].

The specific internal configuration allows selecting channels until 380 bands, with a bandwidth of 1nm. The number of bands that is possible to collect is strictly related to the internal memory of the camera. Thus, in standalone mode (on the flight) is not possible to acquire until 100 bands. In details, the internal configuration includes two sensors, a beam splitter prism and an interferometer. As shown in Figure 15, the Sensor 1 acquires near-infrared bands, from 659.2 nm to 802.6 nm, the Sensor 2 captures visible bands, from 502.8 nm to 635.1 nm. The gap among 659 nm to 635.1 produces a discontinuity in the spectral profiles. The CMOS sensors have both an aspect ratio of 1:1 with photodiode of 5.5 microns.

The interferometer is a tunable optical filter, composed of two partially reflective parallel plates with variable distance (air gap), controlled by piezoelectric actuators [181], [183], [184]. The interferometer is based thus of the principle of dichroism. The selection of the specific wavelengths is controlled by the FPI gap (air-gap). The air gap, indeed, produces many refractions and reflections, and the peak of wavelengths that passes through the second plates. Equation (4) defines the relationships among the air gaps and the rise of wavelengths:

$$\lambda_{peak,m} = \frac{2dn}{m} \tag{4}$$

where:

λ is the wavelength subjected to resonance;
d is the air-gap;
n is the refraction index of the interferometer;
m is the order of the peak.

The possibility to select the air gap, for this camera, allows to control also the bandwidth. Therefore, it is possible to acquire the same wavelength with two different values of FWHM changing the distance between the interferometer plates. The bandwidth can assume values from 18 nm to 10 nm for wider gaps. The software of the camera allows to use the camera with two FWHM setting: *narrow* o *wide*. Narrow option for FWHM refers to that higher gap index wavelengths, while the wide choice to the lower. The acquisition in narrow mode, thus the decreasing of the band with, allows higher spectral resolution but produces different discontinuities in the spectral profile. The effects of the use of narrow or wide setting are shown in Figure 16.

The incident radiation passes through the optical assembly, and then through the FPI interferometer. In the end, a beam splitter prism redirects the radiation to two CMOS sensors. The presence of the interferometer and two not aligned sensors produces a slight variation of the optical path that, therefore, makes the use of this sensor challenge.

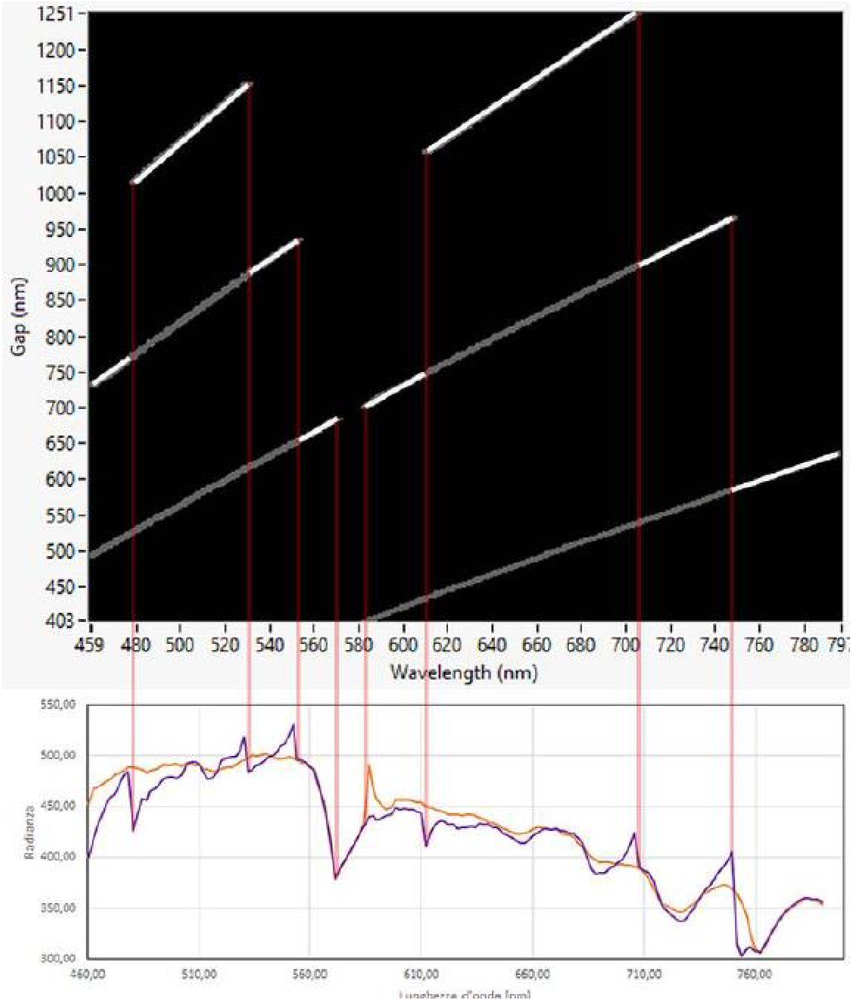


Figure 16. The wavelength/gap graph extracted by the HSI software presents the branch of wavelength sequence. The spectral profile, measured with an acquisition step of 2 nm, indicates the effects of the narrow or wide FWHM camera setting. In purple profile, the radiance (mW sr^{-1} nm^{-1} m^{-2}) in narrow sequence, in orange, the wide one [185].

3.2.2 Radiometric calibration and inter-calibration

As mentioned before the spectral calibration and the radiometric sensor characterization procedure, for the analysed camera are carried out by the manufacturer. Thus, they are out of this dissertation. For what concern the other steps of the radiometric calibration, the users can manage them during the survey or in post-processing. Figure 17 summarizes the radiometric calibration process for the Rikola camera, pointing out the main necessary tools and software to operate. Indeed, the dark current correction and the vignetting correction can be made through the HSI software of the camera [186], while for the estimation of the reflectance factor, external software such as ENVI software (version 4.7 2009) and the "empirical line tool" [187] have to be used.

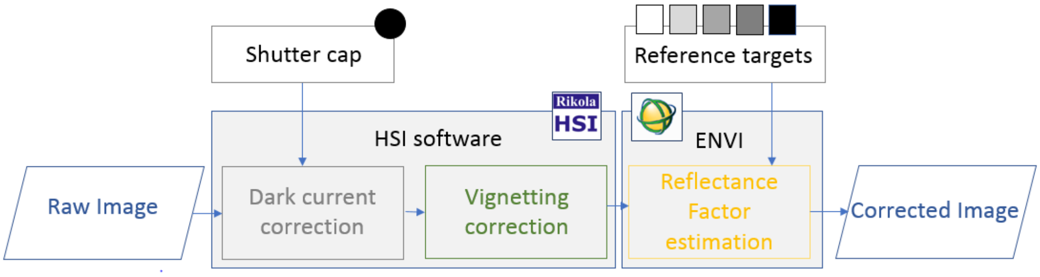


Figure 17. The radiometric calibration process for Senop Rikola camera.

From the operational point of view, the dark current can be acquired before each acquisition using a shutter cap. Instead, the vignetting effects are corrected by the software during the "calibration process" integrated. For the estimation of the reflectance factor, five calibrated reference panels are employed. Looking at the calibrated panel used for the analysis of this research, two sets of panels are used based on the GSD of the acquisition. For close-range applications, the MAPIR Reflectance Calibration Ground Target (31.75 × 25.4 cm) with four reflectance values (2%, 21%, 27%, 83%) was used [188]; while for UAV application the EnsoMosicMill Reference targets (50×50 cm) were used (2 %, 9 %, 23 %, 44 % and 75 %)[189] (Figure 18).

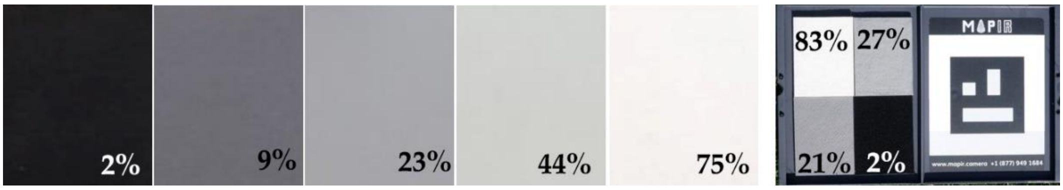


Figure 18. EnsoMosaicMill targets and MAPIR panel.

To be confident that the radiometric characteristics of the camera remain in place, a radiometric inter-calibration was performed in collaboration with CNR IREA Milano. A comparative analysis among the spectral response of the Senop Rikola and the Spectral Evolution SR3500 (25° FOV and spectral resolution of 3nm) was developed. For this purpose, the target acquisitions were performed in the laboratory, with the use of halogen lamp, and outdoor, to test the camera in different conditions (Figure 19). The MAPIR target and the Spectralon white target (94%) were used. The spectroradiometer was used to measure both the irradiance of the Spectralon white and the radiance of the others. The spectral step of 3 nm was set for the measurement of the spectroradiometer.

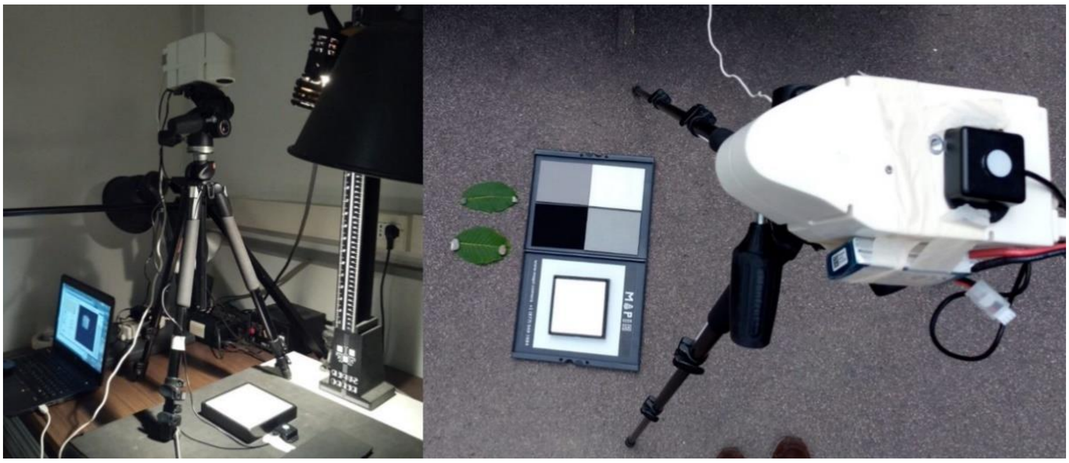


Figure 19. On the left the acquisition set in the laboratory, on the right the outdoor test set.

However, for the Senop Rikola, the spectral step of 3 nm was set with a narrow bandwidth. Moreover, the spectral signatures of two leaves as real surfaces were collected with the two sensors. Figure 20 presents the spectral signatures of the MAPIR black and white panels; instead, Figure 21 the spectral signature of the one leaf, as a sample.

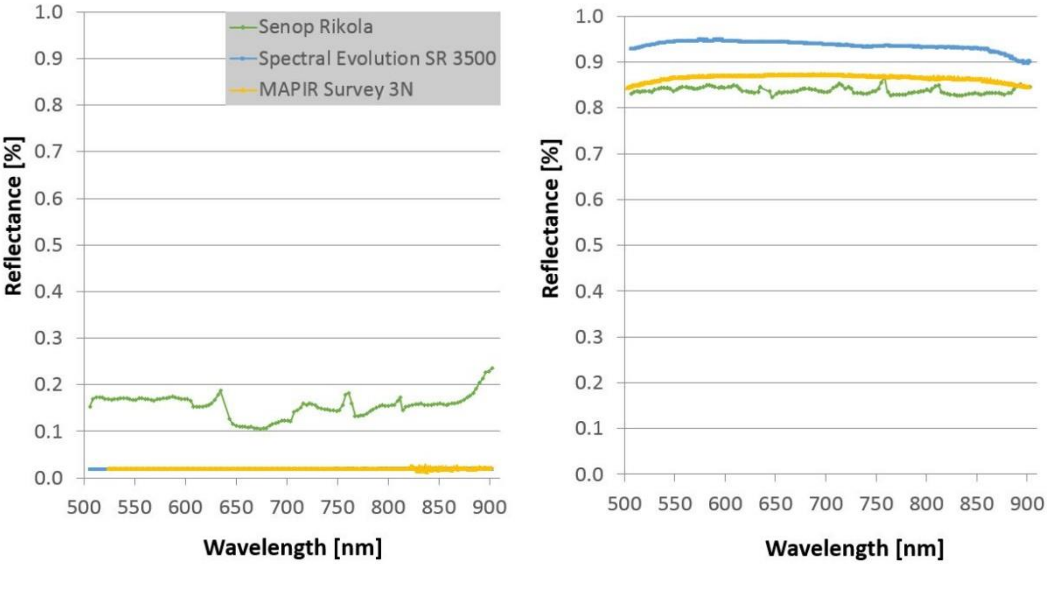


Figure 20. On the left the spectral signatures of the MAPIR black panel, on the right the MAPIR white panel.

As it is possible to notice from Figure 20 and Figure 21, the profiles of the Senop Rikola present slight variations compared with the MAPIR reference curves and the Spectroradiometer measures. This effect is due to the spectral variability of the signature in a region of interest (ROI), that is more evident for low reflectance targets.

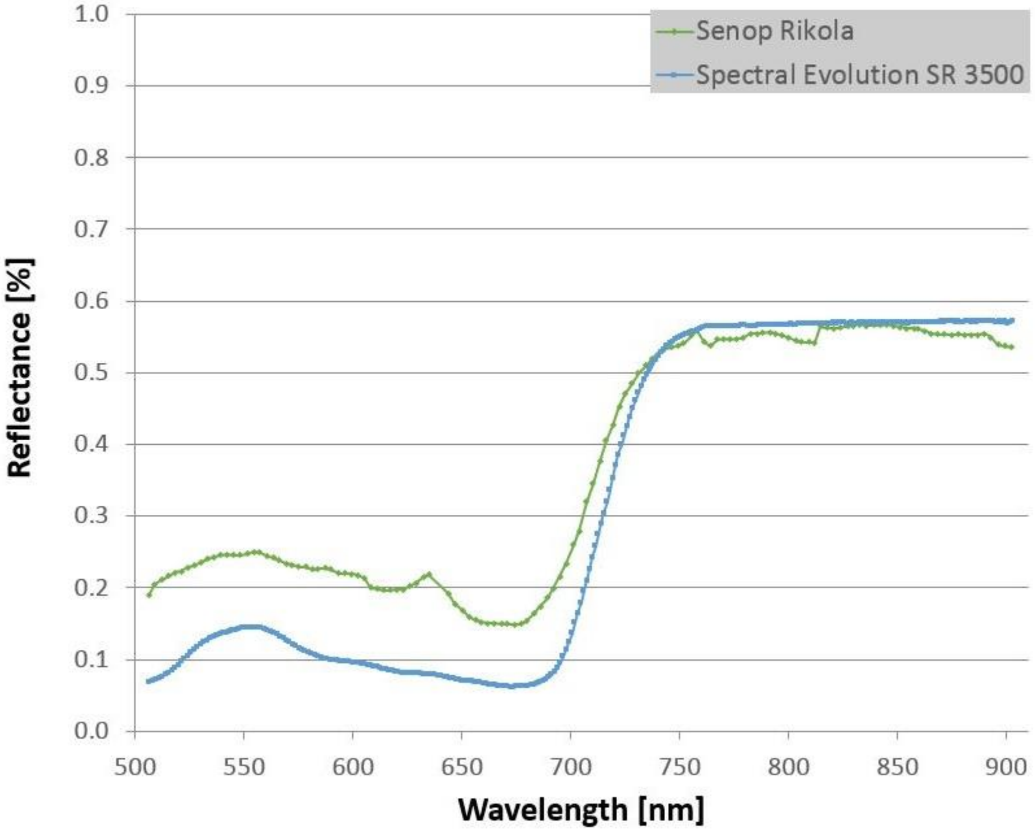


Figure 21. Spectral signatures of the leaf measured by the Senop Rikola (green) and Spectral Evolution SR3500 (blue).

3.2.3 Geometric calibration

As mentioned above, several studies reported the procedure of geometric calibration of the Rikola camera. However, it is impossible to retrieve from them the calibration parameters of each band and most of all; the literature underline the gap to test the stability of these parameters [178]. For this reason, in this work, an in-depth study on the IOPs parameters stability has been performed [190].

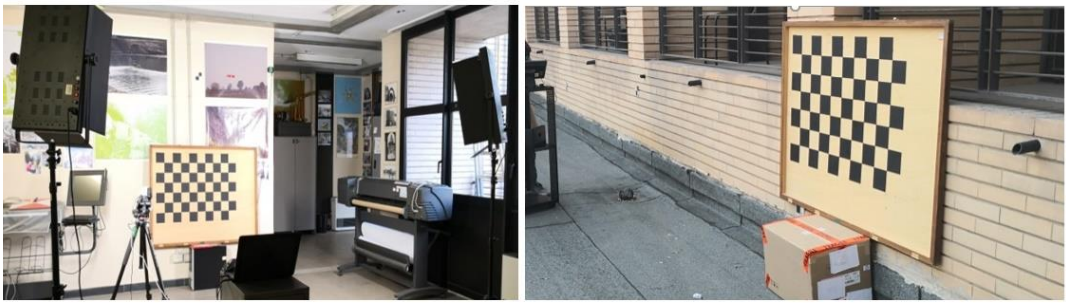


Figure 22. The acquisition set in an indoor environment, considering controlled illumination and exposure values (on the left), and in an outdoor environment (on the right). The calibration panel is a chessboard with an internal array of black and white squares (size of 10 cm).

The reliability of the inner orientation parameters was evaluated in different environmental conditions (indoor with several illumination conditions and outdoor) and different times (Figure 22). Using a self-calibration approach, the proposed methodology was based on the steps illustrated in Figure 23. For the geometric calibration procedure, the HSI software, ENVI and Matlab® were used.

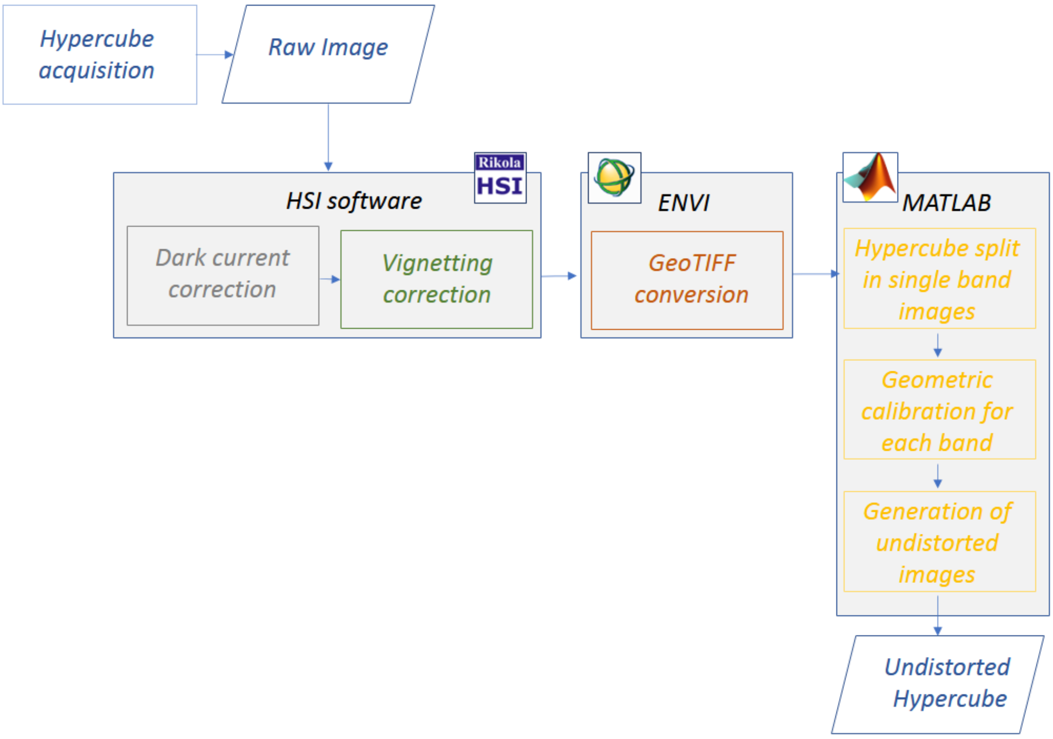


Figure 23. Geometric calibration workflow.

In details, the hypercube acquisition was designed according to close-range photogrammetry rules for the SfM acquisitions. For the camera, the same setting reported in the literature was chosen in the first place to check if the parameters were the same. Thus, image resolution (1010×1010 pixels), spectral range (502-906 nm), spectral resolution (12 nm), wide option, 24 bands were chosen. Instead, for each test, the integration time was set according to the illumination and the environmental conditions. Table 10 summarizes the main features of each test.

Table 10. Geometric calibration. Test configurations.

Test Number	Environmental conditions	Integration time (ms)
T1	Indoor uncontrolled	500
T2	Indoor controlled	1000
T3	Outdoor	10

The cubes were collected in .bsq format by the camera and then converted in GeoTIFF images using the ENVI software (version 4.7 2009). A customized routine in Matlab® was developed to split the datacubes in single bands (Appendix A) and estimate the inner orientation parameters for each channel (Appendix B). The process allowed to estimate the coordinates of the principal point, the focal length, the radial distortion coefficients, and the tangential distortions.

Some considerations on the calibration parameter are presented. It is worth to underline that two groups of calibration parameters characterize the camera having two sensors.

Figure 24 shows the values of the focal length (c_x, c_y) for each band. It is possible to notice there two clusters of values, one for each sensor. Moreover, the focal length change during the time, with consistent differences between Test 1 (the red) and Test 3 (green).

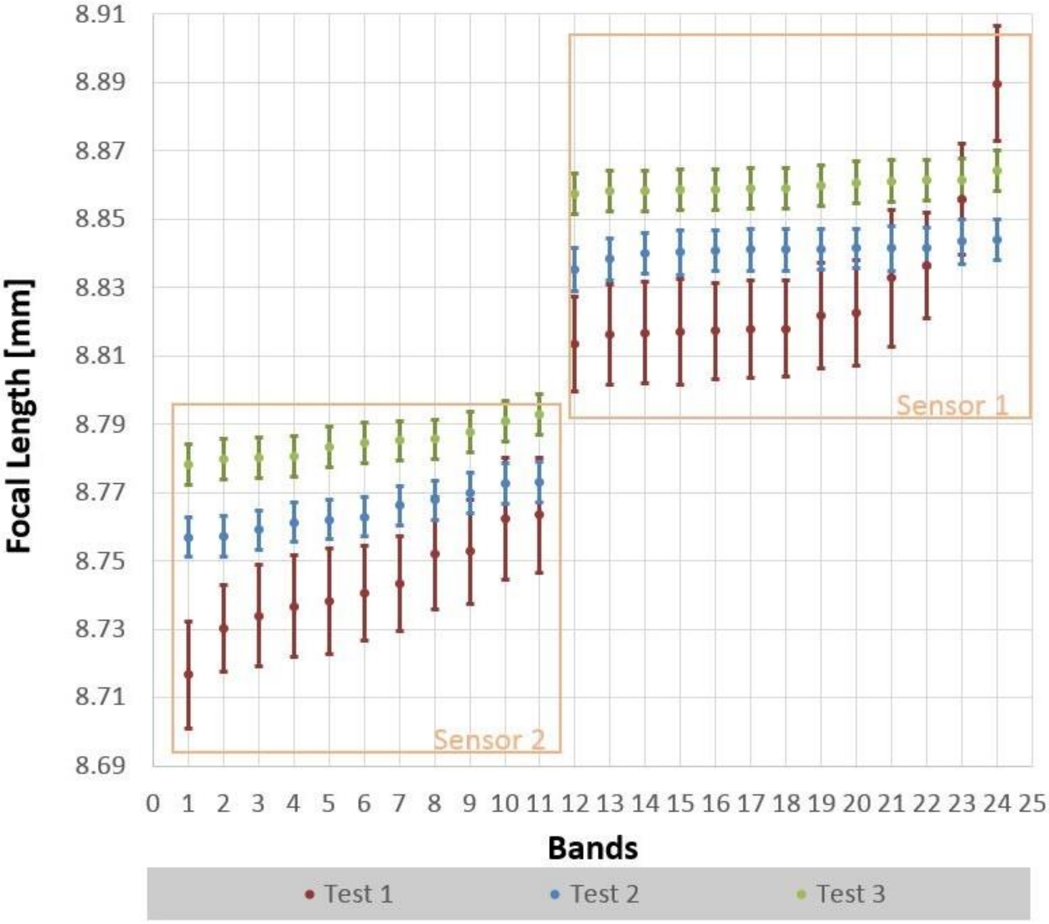


Figure 24. Focal length distribution for each spectral band.

Figure 25 illustrates the principal point coordinates (x_0,y_0) for each configuration and each sensor. It is possible to notice that the principal point position is stable for each test. However, the results of Test 1 are very different from the other two tests. The two sensors also present for principal point different values, but the discrepancies, in this case, are less evident than in the case of the focal length.

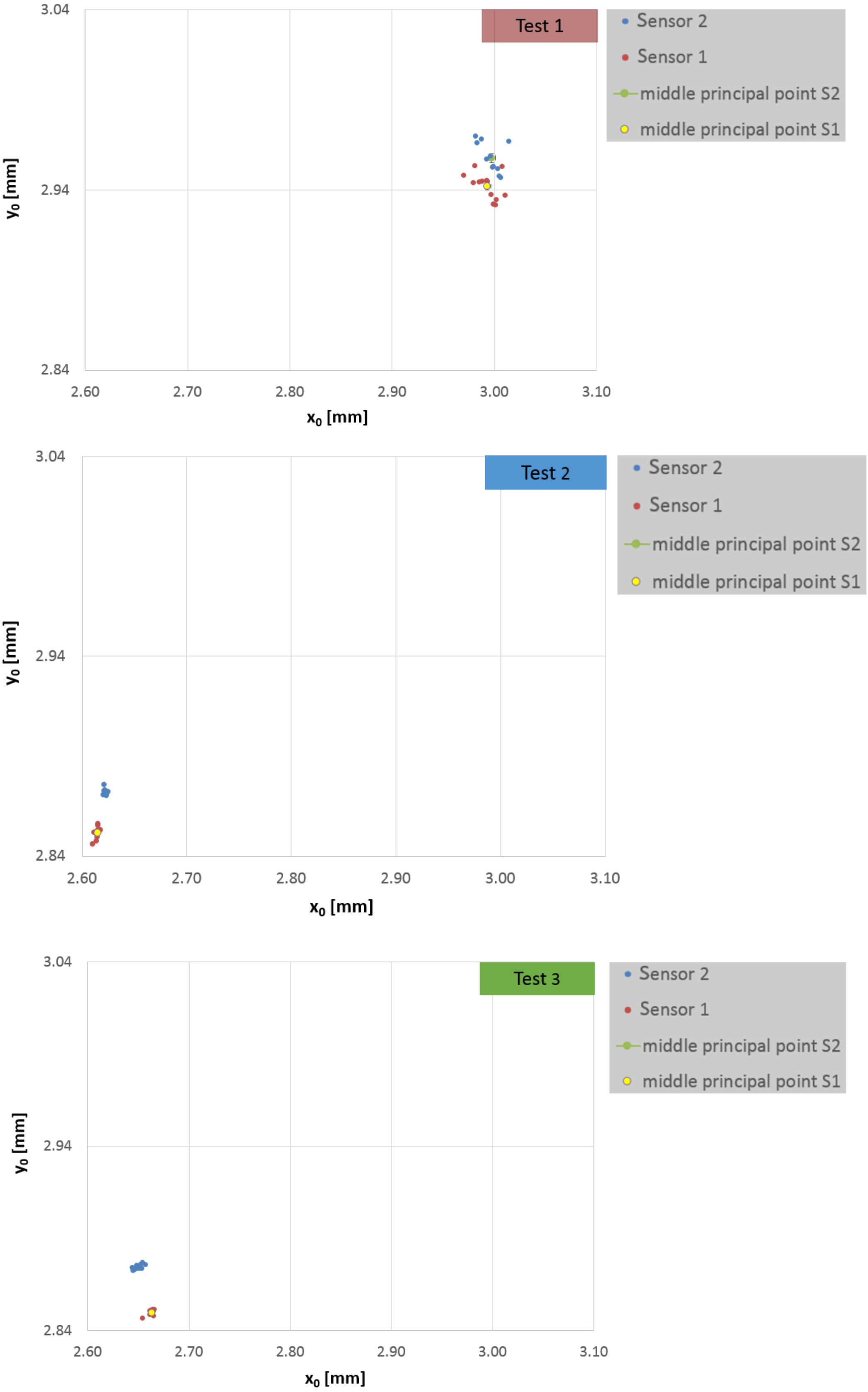


Figure 25. Principal point coordinate distribution in T1, T2, and T3.

Table 11 reports the medium values of focal length (*c*), the principal point coordinates (x_0,y_0) estimated during the different tests of this calibration procedure and the values reported in the literature (*L*) [108]. As mentioned before, it is impossible to retrieve the precise calibration parameters for each band from the literature. Thus, the average values are compared. It is possible to notice that the difference from the parameter calculated with this procedure exists.

Table 11. Comparative analysis of the average calibration parameters, in details, focal length and principal point coordinates, with their standard deviations ($\overline{\sigma}$). L refers to the calibration parameters reported in [108] (in grey), T1, T2, T3 were the calibration parameter extracted from the calibration procedure.

	Sensor	c [mm]	$\overline{\sigma}_c$ [mm]	x_0 [mm]	$\overline{\sigma}$ x_0 [mm]	y_0 [mm]	$\overline{\sigma}$ y_0 [mm]
L	Sensor 1	8.7000	± 0.005	3.1805	±0.0043	3.1884	±0.0022
	Sensor 2	8.6556	± 0.010	3.1399	±0.0142	3.1929	± 0.0046
T1	Sensor 1	8.8289	±0.0003	2.998	±0.0031	2.942	±0.0019
	Sensor 2	8.7428	±0.0003	2.993	±0.0024	2.958	±0.0029
T2	Sensor 1	8.8408	±0.0001	2.6150	±0.0005	2.8516	±0.0007
	Sensor 2	8.7645	±0.0001	2.6227	±0.0004	2.8717	±0.0004
T3	Sensor 1	8.8599	±0.0001	2.664	±0.0007	2.850	±0.004
	Sensor 2	8.7845	±0.0001	2.651	±0.0004	2.874	±0.0012

For what concerns the radial and the tangential distortions, Table 12 reports the coefficients estimated for each test and each sensor. Looking at the values of both the radial and the tangential distortions, it is possible to notice that the tangential distortion coefficients (P_1 and P_2) are negligible compared with the radial ones. Figure 26 shows that the radial distortions have a *barrel* shape, and they are more evident on the border of the image. Looking at the variation of the radial distortion coefficients in the three tests, it is possible to underline that the differences are minimal. The graph of the radial distortion differences (Figure 26) illustrates that the radial distortion differences are less than one pixel.

Table 12. Tangential distortions coefficients for T1, T2, and T3.

	Sensor	k_1 [pixel^{-2}]	k_2 [pixel^{-4}]	k_3 [pixel^{-6}]	P_1 [pixel^{-1}]	P_2[pixel^{-1}]
T1	Sensor 1	-0.3160	-0.0046	1.3774	-0.0001	-0.0001
	Sensor 2	-0.3247	-0.1165	2.1300	-0.0009	-0.0014
T2	Sensor 1	-0.3165	0.3537	-1.1535	-0.0010	-0.0015
	Sensor 2	-0.3250	0.3164	-1.0263	-0.0009	-0.0014
T3	Sensor 1	-0.2950	0.0011	0.5362	-0.0001	-0.0007
	Sensor 2	-0.3094	0.1101	-0.2770	-0.0000	-0.0006

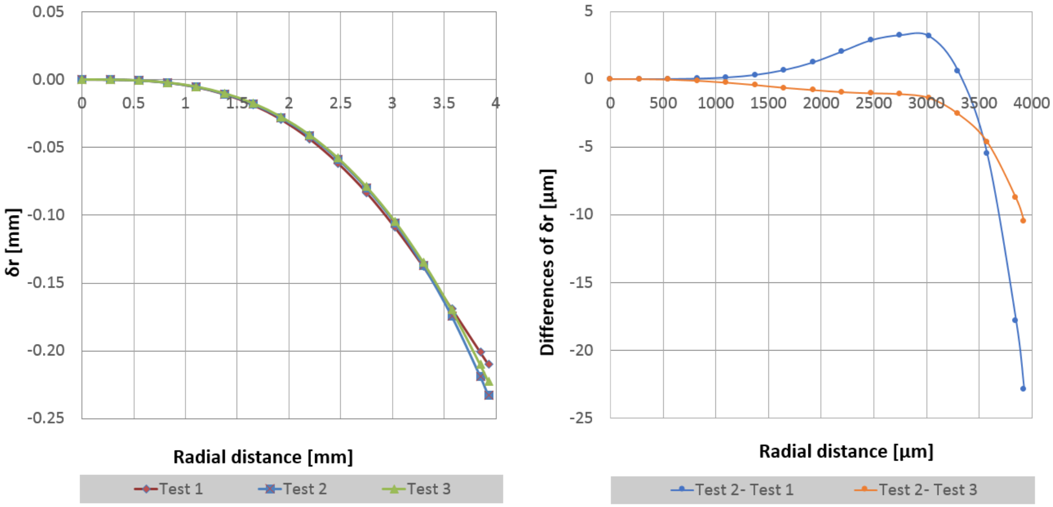


Figure 26. Radial Distortion Curves for T1, T2 and T3 (on the left), and the differences of radial distortion among tests (on the right).

Further analysis was performed to evaluate the impact of the calibration parameter variations on geometric precision. The parameters of Sensor 1 and Test 2 images were used. Thirty-one distorted images were corrected exploiting three sets of calibration parameters, and four scenarios were analyzed according to Figure 27:

(1) the nominal parameters which mean the focal length provided by the producer (9 mm);
(2) the MatLab calibration obtained by the procedure described above;
(3) the adjusted parameters calculated by Agisoft Metashape professional.

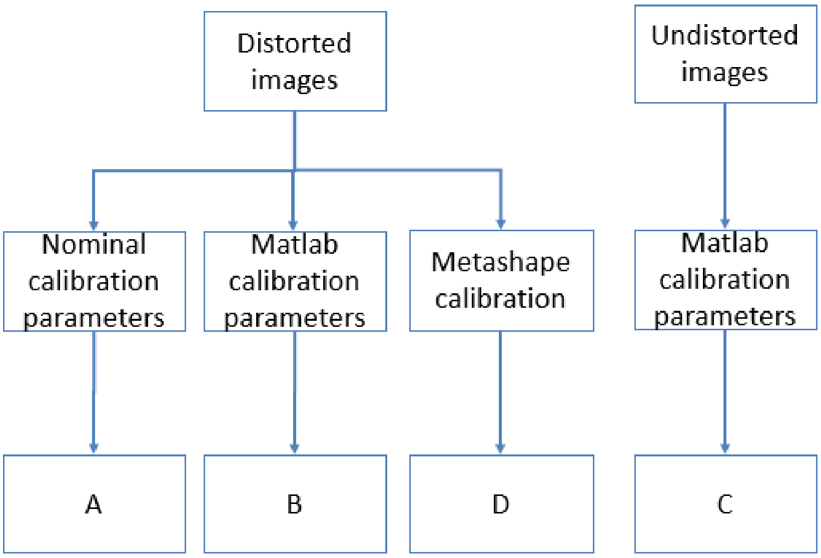

Distorted images
Undistorted images
Nominal calibration parameters
Matlab calibration parameters
Metashape calibration
Matlab calibration parameters
A
B
D
C

Figure 27. Evaluation of the impact of calibration parameters on the geometric precision. The four scenarios.

To the best of the authors' knowledge, Agisoft Metashape and similar software are not yet able to elaborate spectral data with more than four bands as-is, but it is necessary processing the channels one by one. Thus, A single band analysis was being performed. Five checkpoints, as shown in Figure 28, were be selected with a multiplicity of 31. The Fraser's formula was used for comparing the estimation precision of points coordinate into the calibrated images (eq. (5)) [191].

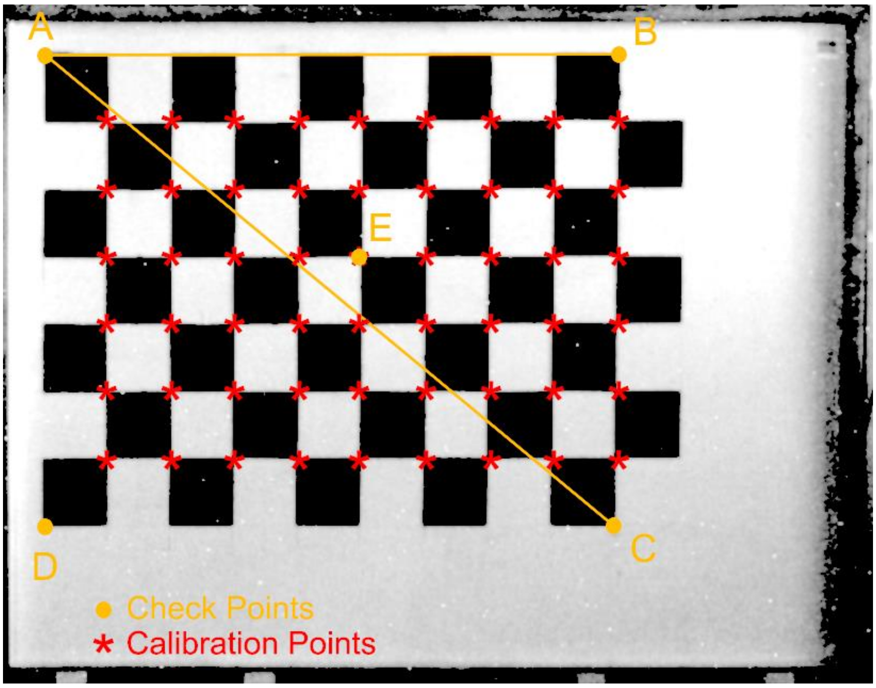


Figure 28. Check points (in orange) chosen for the precision estimation.

$$\sigma_{XYZ} = \frac{qZ}{c\sqrt{k}} \sigma_{P_\xi} = 0.2\,mm \tag{5}$$

where:

q: shape factor that represents the goodness of the configuration of the intersections among several images in the space. It can range between 0.4 and 0.8 for favorable generic convergent configuration, and it is strongly dependent on the operator experience (0.8);

Z: distance from the object (2000 mm);

c: focal length (9 mm);

k: average number of images containing the same point (multiplicity) (31);

$\sigma_{P\xi}$: average error of image coordinates (in this case considered equal to the pixel size: 0.0055 mm);

Table 13 reports the results of the precision estimation. Indeed, comparing the outcomes of the scenarios (A, C, D) and the case (B), it is possible to summarize these general considerations:

(1) the calibration parameters estimated using Matlab improve the accuracy in X and Y;

(2) estimated discrepancies in X and Y are comparable with the nominal precision (σ_{XYZ} = 0.2 mm);

(3) the 2D checkboard calibration panel can cause high discrepancies in Z-direction;

Table 13. Comparative analysis of points coordinate precision into the scenarios summarized in Figure 27. The Δ refers to the differences between the real point coordinates and the estimated ones. The maximum, the average and the minimum difference considering the five points in the 31 images is reported. Scenario B (in grey) is the reference.

Test	Parameter	Δx [mm]	Δy [mm]	Δz [mm]
Distorted – No calibrated (A)	Min	-4.2	-4.7	-0.3
	Max	4.1	3.3	0.9
	med(|Δ|)	3.5	4.2	0.6
Distorted Matlab calibration (B)	Min	-0.9	-0.9	-2.6
	Max	0.5	1.0	-1.0
	med(|Δ|)	0.5	0.5	2.0
Undistorted images (C) - Fixed focal length	Min	-0.6	-0.4	-2.7
	Max	0.9	0.5	-1.6
	med(|Δ|)	0.7	0.4	1.6
Metashape solution (D)	Min	-0.9	-0.9	-2.6
	Max	0.5	1.0	-1.0
	med(|Δ|)	0.5	0.5	2.0

As conclusions related to the geometric calibration of this camera, it is worth to underline that the calibration parameters change in time due to the environmental condition and the internal configuration of the camera. Moreover, the switch-off of the camera cannot guarantee the maintenance of the parameters. To fulfil this gap, the calibration parameters have been estimated using a customized Matlab routine (Appendix A and Appendix B) and evaluated the use of SfM software, such as Agisoft Metashape to perform the self-calibration. The Matlab solution is comparable with the Metashape solution; thus, when the hyperspectral data processing will be fully implemented in SfM commercial or open software, the calibration parameters will also be estimated using them.

3.2.4 Band co-registration

The geometric calibration procedure is not enough to correct the alignment among the bands. Therefore, an alignment process is needed.

Figure 29 illustrates the effects of the band misalignment. As mentioned above, the Matlab routines for the co-registration process of the hyperspectral image are not available. Thus, for this research, a band coregistration routine was customized for hyperspectral images (Appendix C). For the coregistration, an anchor band was used as a reference, i.e., the first band. The routine employs the *imregconfig* function [192]. The function performs intensity-based image registration. It is possible to set two different modalities based on the brightness or the contrast similarities, i.e. *monomodal* or *multimodal*. Monomodal can be used for images with similar brightness or contrast for images acquired with the same sensor.

Multimodal, instead, can be set when the images have different brightness or contrast, or they are collected in diverse exposure configurations. In our case, the "monomodal" parameter was employed according to the type of images. Thanks to an iterative process, the function defines common features, that are used then for the registration. In the end, a 2D affine transformation is applied to the images for concluding the process.

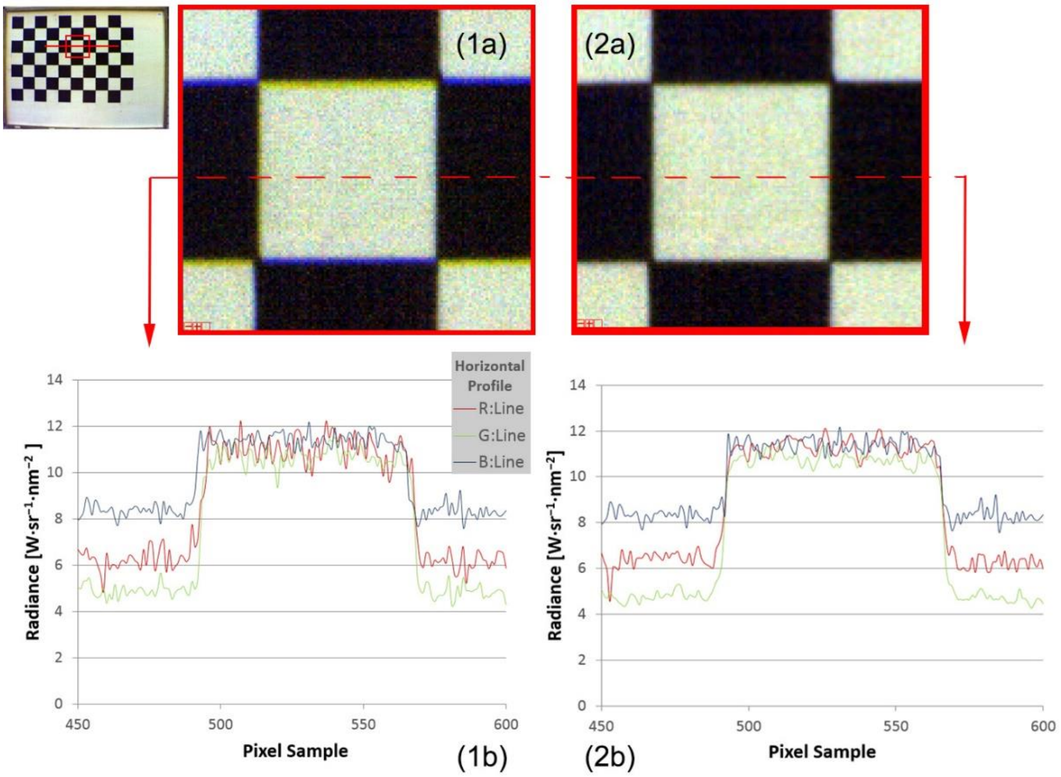


Figure 29. A section of a coregistred hyperspectral image (bands: 24, 16, 1). Effects of band misalignment on the chessboard (1a) and the horizontal profile (1b). The yellow and the blue border around the square unit of the chessboard is due to the misalignment. After the hypercube co-registration (2a) the effect vanishes. The horizontal profile after the coregistration (2b)

Chapter 4

Machine learning for automatic information extraction

After the system configuration and calibration, automatic information extraction techniques, have to be selected, as mentioned in Chapter 1. Information extraction comes from the identification of the relation between the data stored in the remotely sensed image structure and the real target in the scene. The importance of finding a time and cost-effective IE approach of massive amounts of data is evident in the rapid increase of high spatial, spectral and temporal remotely sensed data and emergence in the field of big data.

This process allows the interpretation and the transformation of the data into information, through several steps from feature extraction to scene interpretation (Figure 30).

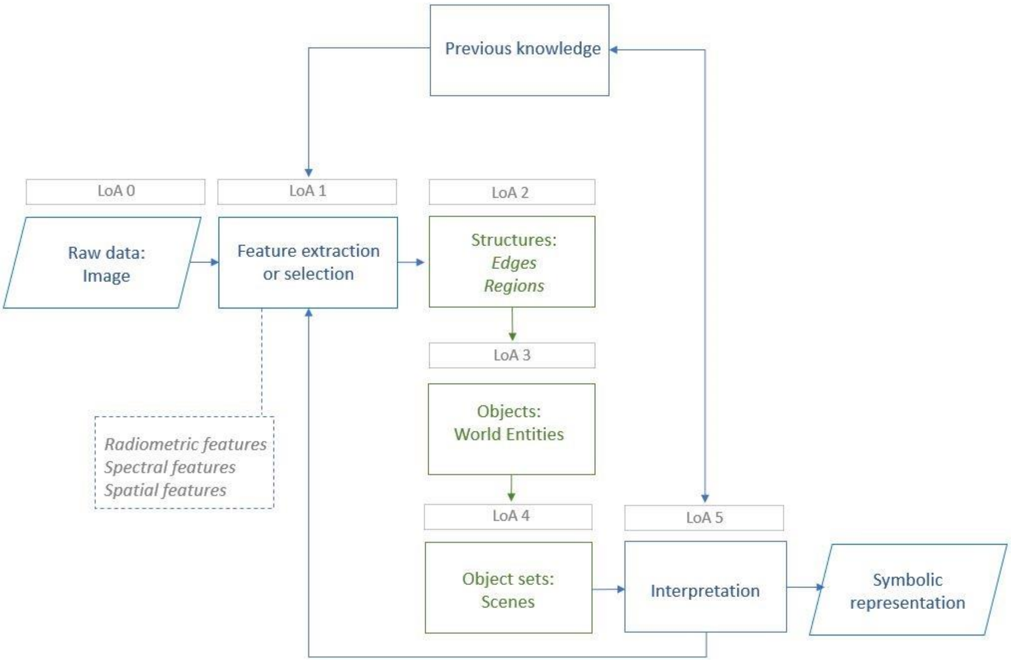


Figure 30. Iterative interpretation process workflow.

The interpretation of remotely sensed images can be an elaborate process to transfer into a computer [193], [194]. For humans, vision is a natural task that can be carried out instantly and automatically. From the study of human perception, it can be learnt that vision is not a straightforward process, because sometimes optical illusions or hidden interpretation layers can affect the process. As in the case of interpretation of handwriting and spoken language, the image analysis has to pass through different information abstraction levels, also known as image interpretation pyramid [195], [196]. The image interpretation *pyramid* allows to describe the scene in a hierarchical way, passing from a symbolic region division to real representation of the object sets.

Moreover, generally a set of image has to be analyzed [197]. Thus, both the object features, and the multiple images imply an iterative nature of the interpretation process and rely heavily on the knowledge and the analyst's expertise.

Time and cost-effective IE approach means in remote sensing thus, automatic processing image techniques able to manipulate a huge amounts of high-resolution (HR) unstructured data, such as images, and to characterize a considerable number of objects by their features to derive information [198]. It is worth to notice that the automatic in this context is referred to the necessity of a small subsample of known information to achieve the full knowledge of the entire image [199].

Several AI algorithms have been developed to aid analysts in image interpretation processes. These algorithms are grouped into a subset called Machine Learning algorithms that can emulate human learning behavior. Indeed, ML is data-driven learning that can automatically learn patterns, to define features, and to assign properties [200]. The features can be classified as following[66]:

- radiometric attributes represent the differences in brightness such as tone (i.e., grey levels of an image region), shadows, and texture (i.e., spatial variations of the tone that are associated with the apparent roughness or smoothness of an object);
- spectral attributes represent the differences in composition or condition that refers to the various combination of the red, green and blue or the spectral signature;
- spatial attributes are the content of a pixel or group of pixels. Example of spatial features can be spatial characteristics of objects such as shape, size or topographic position, regular arrangement of objects (e.g., buildings in residential area or trees in a crop) or association (i.e., the co-occurrence of items) such as playground nearby school or swimming pool associated with recreational centres.

A general strategy for ML to manage High Resolution (HR) data, as presented by Petersson in [200], can be implemented in two different steps: the dimensionality reduction and the interpretation process (e.g., classification or object detection).

This chapter presents the most popular ML approaches for managing HR data (section 4.1) and classify or detect objects (section 4.2). This review aims to show the benefits and the drawback of the most popular algorithms to choose the best fit for the proposed applications.

4.1 Machine learning for automatic feature extraction and selection

IE methods cannot always deal with HR image analysis. Indeed, high resolution refers both to spectral and spatial image characteristics, and it produces a massive and high-dimensionality feature space. The dimension of feature space, which means the input variable or distinctive attribute set, can lead to two main problems. On the one hand, the feature calculation is time-consuming; on the other hand, the IE accuracy can be affected by the 'curse of dimensionality' [201] or 'Hughes phenomenon' [202]. The curse of dimensionality occurs when the increased dimensionality does not boost the IE accuracy [203], [204]. The rise of the sample amount is one of the possible solutions. However, in most applications, definition and collection of the statistically enough sample size requires a strong effort. An alternative solution can be a selection of features through a dimensionality reduction process.

Indeed, based on a review of previous definitions, the features can be classified as (1) strongly relevant, (2) weakly relevant, but not redundant, (3) irrelevant, and (4) redundant [205]. Strongly relevant feature refers to the best feature subset that most contribute to the accuracy. Weakly relevant feature could be useful on certain conditions. Irrelevant features must be discarded. Redundant features are those weakly relevant features that are partially or completely correlated with others.

The dimensionality reduction allows extracting relevant and non-redundant features in a lower-dimensional space, maximizing relevance and minimizing redundancy. In learning tasks, dimensionality reduction leads to the following benefits [206]:

- data collection improvement and optimization;
- limed storage requirement and reduction of computational complexity;
- redundant, irrelevant and noisy data reduction;
- generalization capability increase;
- classification and detection performance increase.

To ensure the optimization of feature subset based on the general classification problem, the dimensionality reduction must consider *2*m - 1* subsets, where *m* is the total number of input attributes. Due to the computational cost, this operation is infeasible; the heuristic methods have been proposed to find a sufficiently good (but not necessarily optimal) subset. Thus, feature subset optimization is a crucial stage for information extraction from high-resolution images.

In a large amount of literature on dimensionality reduction, the reviews represent only a small part [207]–[210]. The most popular methods employed are data binning, feature extraction and feature selection [211]. Each technique has advantages and drawbacks that this section presents, however, in general as pointed out by Bolón-Canedo and Remeseiro in [212], the feature extraction is the one preferred for image analysis in general.

4.1.1 Feature extraction

Feature extraction (FE) refers to a linear or non-linear transformation procedure that reduces the spatial and spectral input space to avoid data redundancy and preserve the most important information [213]. A FE drawback is that the original features are not interpretable and the information about how much an original variable contributes is often lost. However, FE has the significant advantages to create a more compact and more robust discriminating subset and to be a noise-tolerate technique as it helps to avoid overfitting implicitly within learning techniques. Indeed, the application of FE techniques, before undertaking supervised learning, eases the negative effect of the presence of mislabeled instances in the data [213].

FE algorithms are categorized into two main approaches: supervised and unsupervised FE [214]. Unsupervised FE methods do not require any prior knowledge or training data, even though the accuracy optimization in a specific learning task is not their direct scope. Principal Component Analysis and Kernel PCA [215] [216], Partial least Squares Regression (PLSR) [217], Independent Component Analysis (ICA) [213], [218], are some examples. Supervised FE directly considers the available training information for a specific learning task. This category includes algorithms such as Linear Discriminant Analysis (LDA) [56] [219], and Support Vector Machine [219], and its modified version such as and PCA-SVM [220] and LDA-SVM.

In general, it is possible to express the class separability problem for the feature extraction as follows (6):

$$\max_{\vec{w}} J\ (\overrightarrow{w}) \tag{6}$$

where $J(\vec{w})$ is a criterion of the separability measured between classes and $\vec{w}$ is a transformation, e.g., linear projection.

This transformation projects the raw feature space to lower-dimensional space. Thus, feature extraction aims to find a vector $\vec{w}$, known as a linear predictor, that maximizes the class separability [56] [219].

4.1.2 Feature selection

Feature selection (FS) refers to a process to choose a subset of relevant features without any transformations [221]. The whole approach is based on achieving the lowest error in the *learning process* according to a loss functional [222]. FS can be applied directly to the observations, in both the spectral and the spatial domains, or to further extracted features derived from FE. The advantages of the FS method are the preservation of physical meaning and the original semantics, which makes the outputs easy to interpret. Nonetheless, if a small feature set is required and original features are very diverse, information lost can occur because the reduction process leads to the omission of some features [213]. As indicated for the feature extraction, feature selection requires the optimization of an objective function, *J*. The best subset *X* is the results of the following (7):

$$\max_{X \in X_d} J(X) \tag{7}$$

where X_d is the set of all possible subset combinations obtained from the input variables.

In general, there are different ways to gather FS techniques. However, the most common is the one that splits the methodologies into two classes: classifier-independent such as filter methods and classifier-dependent such as a wrapper, and embedded methods [212], [223]. Figure 31 presents the conceptual schema of the cited techniques [206], [224].

In details, filter algorithms remove features directly from the original feature set, without the influence of any machine learning algorithms. Filter methods are based on heuristic ranking criterion (relevance index) that assess the feature relevance depending on the data statistics such as distance, separability, correlation and mutual information. They can rank individual features or the entire subset. Information Divergence Band Selection (IDBS), Constrained Band Selection (CBS), Linearly Constraint Minimum Variance (LCMV), Maximum-Variance PCA (MVPCA) are some example of algorithms that use ranking. Optimal Index Factor (OIF), Maximum Ellipsoid Volume (MEV), Maximum Information (MI), Minimum Dependent Information (MDI), Linear-Prediction-based Band Selection (LPBS), Manifold Ranking (MR), Volume-Gradient-based Band Selection (VGBS) use, instead, the correlation for defining the optimal feature combination [225]. Filtering techniques are fast, scalable and do not involve training the models, and they can handle directly the noise (that in the classification bring to misclassification or mislabeling). On the other hand, they can have some difficulties to differentiate noise from outliers (exceptions) without the expert intervention and to define the expected level of noise required as starting input [213].

Wrapper methods select the feature subsets based on the classification performance (accuracy or error rate) on test data generated by statistical resampling or cross-validation. In this case, a search procedure in the feature space is defined, and various subsets of features are generated and evaluated. The evaluation process implicates the training and testing of the specific machine learning model [226]. This implies as drawbacks that the method is computationally expensive and classifier- specific. Moreover, high risk of overfitting can occur related to the low generalization capability. Despite the filtering techniques that can fail the optimal subset selection, the wrapper methods always provide the best feature set. Some examples of wrapper techniques are Sequential Backward Selection (SBS), Sequential Floating Forward Selection (SFFS), Genetic Algorithms (GAs) [226].

Embedded methods combine the advantages of the filter and wrapped techniques and incorporate the feature selection process into a learning model [224]. Compared to the wrapper method, it has a lower computational cost. It uses the independent criteria to decide the optimal subset for a known cardinality. Standard embedded methods include Classification And Regression Tree (CART), C4.5, RF [225].

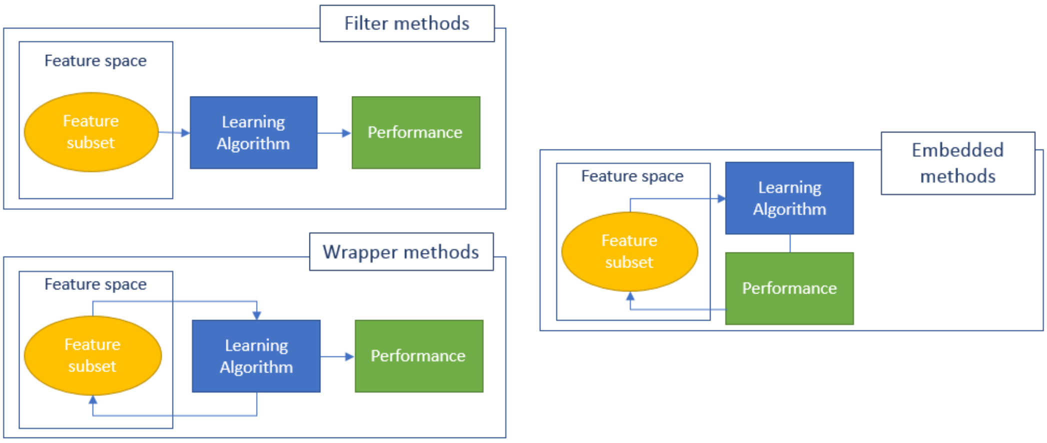


Figure 31. Conceptual schema of feature selection methods.

To conclude this review on the dimensionality reduction, a specific space must be reserved for a problem known as band selection. The band selection concerns process to discriminate significative wavelengths from HSI.

The problem, in this case, is the high correlation among adjacent bands and the lack of discriminative information in some of them. Sun and Dun in [227] provided an extensive review of the most popular methods explicitly used for band selection. This review points out that it is not yet possible to identify criteria for the appropriate defining size of the band subset and choosing the best band selection; however, feature extraction performs better for larger hyperspectral, compared to the feature selection that is more suitable for smaller dataset due to the computational time cost. Additionally, when dealing with hyperspectral noisy data, the PCA-based criteria performs better than others in real-world applications [228]. Indeed, the Principal Component Analysis is the baseline algorithm because it can perform feature extraction as well as feature selection [56], [229]. PCA, also called Karhunen-Loeve or K-L method, is defined as an orthogonal linear projection of the original data space onto a new smaller space with uncorrelated features, i.e., principal components (PCs).

In the first version of PCA, the features are constrained to be linear functions of the input variables, however, non-linear implementation of the PCA are available [230]. In this section, the focus is on the linear PCA to point out the basics and the main principles that allow using this algorithm both for feature extraction and selection.

The procedure follows these steps [231]–[233]:

(1) The input data are normalized for including all the attribute in the same range. This step guarantee that attributes with large domains will not dominate on the smaller.

(2) PCA computes *K* orthonormal vectors, i.e. principal components. These are unit vectors that each point in a direction perpendicular to the others. The input data are a linear combination of the principal components. The (8) represents the matrix form of the PCs:

$$Y_i = \begin{pmatrix} w_{1,1} & \cdots & w_{1,j} \\ \cdots & \cdots & \cdots \\ w_{i,1} & \cdots & w_{i,j} \end{pmatrix} X_j \tag{8}$$

where:

Y is the PC vector;

W the transformation matrix;

X the vector of the original data. The coefficients of the transformation matrix *W* are the eigenvectors that diagonalize the covariance matrix of the original data.

(3) The principal components are sorted in order of decreasing *significance* according to the *W* transformation matrix. The equation can be used (9) to evaluate the contribution of the new features:

$$W_k(i) = \frac{P_k\ (i)^2}{\sum_{i=1}^{n} P_k\ (i)^2}\,, \qquad i \in \{1, \dots, n\} \tag{9}$$

where:

$W_k(i)$ is a weighting factor of the i-th input variable of the *k-th* principal component;

$P_k\ (i)$, i.e. the loading vectors. The weightings of the first several principal components can be examined to select the significative feature set.

There are three practical criteria to select the most representative PCs [234]:

- Cumulative percentage of total variation: the representative PCs must contribute to the cumulative percentage of total variation to achieve 80% or 90%.
- Kaiser's criterion: the selected PCs must have a variance that exceeds 1, which means that if all the variables are independent, the PCs have variance equal to 1 in the correlation matrix.
- Screen graph: it is the plot of eigenvalues in decreasing order. The PC number can be selected looking at the *elbow* in the graph, that indicates the components to retain. The trend of the function must be sharply decreasing on the left of *elbow* point, while on the right, it has to be constant or weakly decreasing [235].

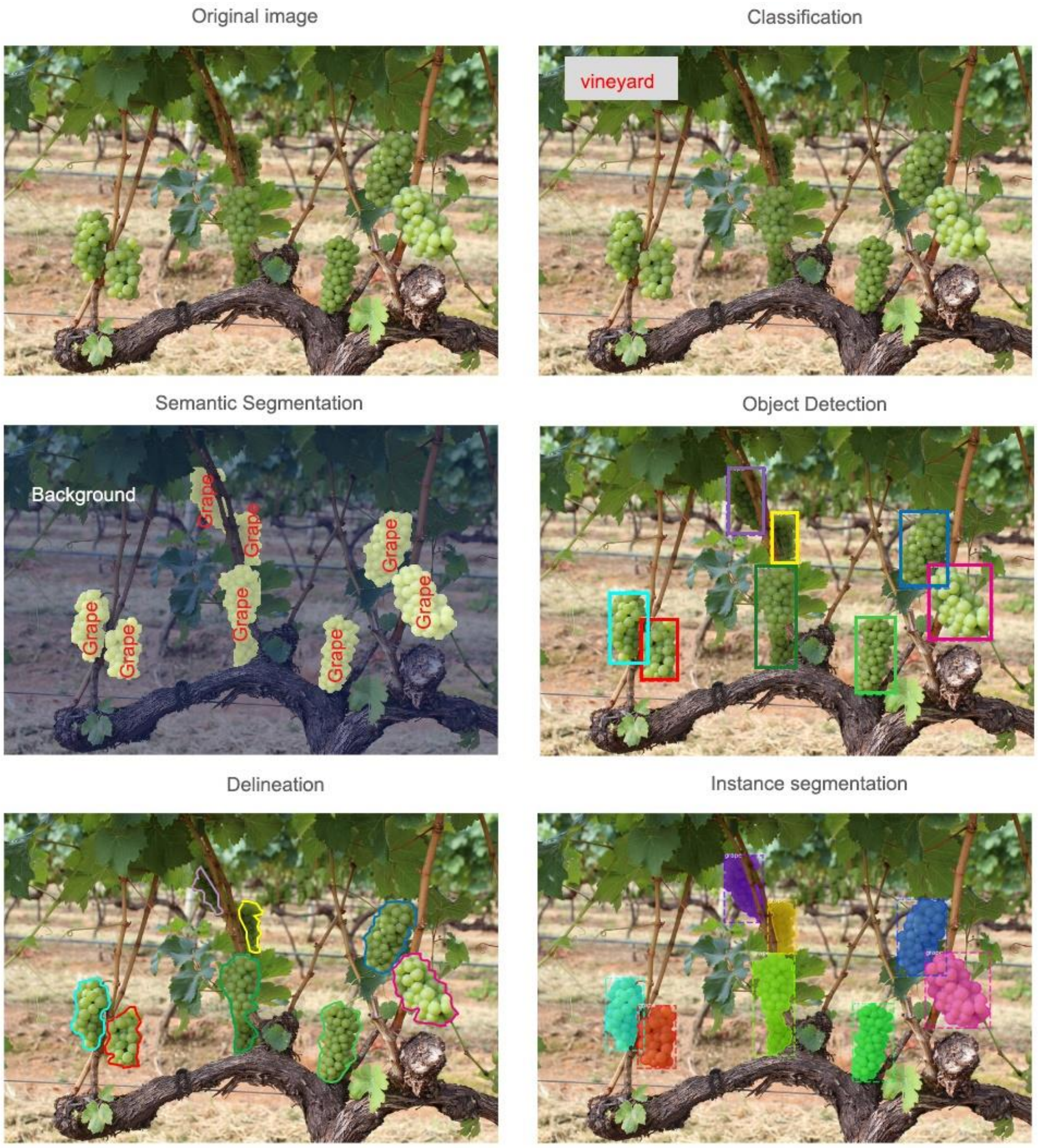
Once the PCs have been chosen, the interpretation of them is based on eigenvectors. The meaning of PCs can be determined looking at the coefficient ($w_{i,j}$) of variables X_j. The greater $w_{i,j}$ is, the higher the correlation, and X_j is the most important for the PC [236].

4.2 Machine Learning for automatic information extraction

The Information Extraction for high spectral and spatial resolution remote sensing images can be accomplished through different image interpretation tasks [237], [238]:

- segmentation: the division of an image into connected regions or categories which correspond to objects or parts of objects [239];
- classification or pattern recognition: the category recognition, i.e. the identification of the class associated with objects;
- regression: the classification in continuous numerical value;
- object detection: object localization which means to separate objects of interest from the background in a target image;
- instance segmentation: category recognition, localization and single items list;
- enumeration: list or counting discrete items visible on an image;
- mensuration: measurement of objects and features in terms of distance, height, volume, or area;
- delineation: drawing boundaries around distinct regions of the image characterized by specific tones or textures.

The output of the cited task can be initially in raster format, but they may be generalized to polygons with further processing. The Figure 32 shows the output of each task [240].



Original image

Classification

Semantic Segmentation

Object Detection

Delineation

Instance segmentation

Figure 32. Image interpretation tasks.

To achieve the different tasks, in literature, it is possible to find numerous techniques that can be categorized according to different taxonomies [241], [242]. The main taxonomies can classify the algorithms based on the following rules:

(1) basic unit;
(2) degree of automation;
(3) assumption of on data distribution.

According to the basic unit, ML algorithms can be classified as a pixel, subpixel and object-based (spatial-based) methods.

Pixel-based approaches use pixels as the base units. The pixel can be a limit because they can include multiple objects with different features at such remote sensing scale [94]. However, within these numerical limits, per-pixel learning can be consistently high if the appropriate spatial resolution (i.e., pixel size) is defined according to the suitable LoD. Some per-pixel techniques are focused exclusively on maximizing computational class separability, using the traditional maximum likelihood algorithm and support vector machines, while others developed methodologies that imported extraneous information when aggregating spectrally similar pixels, by incorporating contextual relationships or by measuring pixel interconnectivity.

The subpixel technique overcomes the pixel constraint of the per-pixel classifications to statistically measure spectral mixtures representing multiple classes within individual pixels [243]. Among the algorithms, linear spectral mixture analysis, regression analysis, and regression trees have had a broader appeal because they are theoretically and computationally simpler, as well as more prevalent in many commercial software packages. The main limitations of using these techniques are related to the identification of spectrally pure endmembers, preferably using reference samples collected in the field. Nevertheless, Weng and Hu (2008) [244] discovered that Artificial neural networks (ANNs) were also capable of performing non-linear mixing at the subpixel level in remote sensing applications. In response, a Multiple Endmember Spectral Mixture Analysis (MESMA) has been developed to identify many more endmember types to represent the heterogeneous mixture [245].

Object-based Approaches (OBIA), also called Geographic Object-Based Approaches (GEOBIA), split an image into regions (objects) of different sizes containing multiple pixels based on spectral similarity and spatial proximity [211]. The object-based methods differ from pixel-wise techniques because they incorporate in the learning process the spatial features. The object-based methods suggest a two-stage algorithm. In the first step, pixels are merged to object clusters, possibly in a multi-level object hierarchy, which then will be analyzed and classified in the second step [246]. SVM approach [247] or K-Nearest Neighbors (K-NN) algorithm [248] can be cited as examples.

According to the degree of automation, ML algorithms can be classified based on the degree of automation and thus the amount of human interaction in unsupervised, supervised, and Reinforcement Learning (RL).

The unsupervised approach does not require guidance from the analyst before processing, and they are applied when prior knowledge of the ground-truth is not available. The algorithm role is the definition of the relationship between feature space (pattern) and information classes (statistical representative clusters). The algorithm operates under constraints specified by the user such as cluster number, spectral and spatial search radius, bands used, and iterations defined for producing the output map. The image analyst, in a second step, must determine if these arbitrary classes have meaning in the context of the end-user application. In this case, experimentation determines the optimal number of unique categories used for initialization of the algorithm. Furthermore, classes may not be transferrable to another image. Unsupervised learning includes principal component analysis, and generative adversarial networks (GANs), convolutional neural networks (CNN).

Supervised learning implies the image analyst support, who has to define an appropriate classification scheme (class definition) and then identify most representative training data in the imagery for each class. The training data allow learning the rule for assigning labels outside the training areas. In general, training data are not transferrable for a scene with different conditions (e.g. lighting conditions, or atmospheric effects) and across time. K-nearest neighbors (KNN) [249], SVM, random forest (RF) and artificial neural network (ANN) and Deep Neural Network (DNN) [250] are widely used nowadays since they show better performances in various classification and regression problems.

Reinforcement learning is an adaptive approach for learning through trial-and-error interaction with a dynamic environment [251], [252] [253]. However, these methods are limited only to simple applications, e.g., image cropping and image enhancement [254]. In literature, only a few applications in the field of image analysis tasks, such as object detection or segmentation, are available [254], [255].

Finally, according to the assumption of data distribution, ML algorithms can be classified as parametric and non-parametric approaches [256]. The parametric learning, assuming that the data are normally distributed, summarizes data with a set of parameters of fixed size (independent of the number of training examples). In other words, they are based on the statistical probability distribution of each class. Linear Discriminant Analysis and maximum likelihood are some examples.

Non-parametric approaches, instead, does not require normally distributed data. This makes these types of algorithms more suitable for real applications where data are not often normally distributed. k-nearest neighbors, Support Vector Machines or decision tree are some examples.

As explained by Jensen (2005) [66], *no learning method is inherently superior to any other*. Problem task, the study area, the data sources, the intended use of the results, the LoD analysis are the main parameters for choosing the most appropriate efficient, timely and cost-effective approach. In particular, scale appropriateness, thus the spatial resolution of the data and the LoD, has an important role in the selection of the learning methods. There is no universally accepted method to determine an optimal level of scale (e.g., object size) to detect objects, and a single scale may not be suitable for all classes. It is worth to underline that the accuracy of the supervised approach is generally higher than the unsupervised, especially in a complex context where spectral/temporal manifestation of different categories may be similar.

The following sections presents the most representative algorithms for semantic segmentation (section 4.2.1) and object detection (section 4.2.2) that fit with the study case of this thesis.

4.2.1 Semantic segmentation

As introduced above, the aim of semantic segmentation is the division of the image in different ROIs, assigning to each pixel a category label from a pre-defined set [56], [257].

Among the algorithms reported in the literature [258], Support vector machine and Random Forest are becoming increasingly popular in remote sensing applications and the analysis of hyperspectral data [259]. The main reasons are related to their excellent capacity to manage the noise and the overtraining and the unbalanced dataset. To accomplish semantic segmentation also Liu et al. [260] also tested Fully Convolutional Neural Network (FCNN), but the experiment analysis demonstrated any advantages of using this technique the case of few training samples are available, as happen in most of the remote sensing applications. Due to this consideration, in this section, a brief description of the main feature of only conventional techniques are presented, including the principal metrics for evaluating the performance.

Support vector machine is formulated for a binary problem and can be employed for the semantic segmentation task [42]. As reported in [241], SVM was applied in remote sensing field for the first time by Gualtieri and Cromp [261], however, a more recent treatment is provided in [262]. According to the literature, the main advantages to using the SVM on high-resolution data (especially HSI), in the first place, are its demonstrated effectiveness in the analysis of hyperdimensional data also without any feature-reduction and its good generalization capability. Moreover, it works well in the presence of a few training sample and heterogeneous features. The low sensitivity to the number of training samples depends on the fact that SVMs implement a classification strategy that exploits a margin-based geometrical criterion rather than a purely statistical measure.

The SVM drawback is that it was initially designed to solve binary problems. This drawback becomes even more evident when dealing with HSI because the spectral similarity among some classes.

SVM can be employed also in the multiclass problem through two different techniques: parallel approaches or hierarchical tree-based approaches. The parallel methods can be implemented according to two different strategies: the One-Against-One (OAO) or the One-Against-All (OAA). The first approach, one-against-one, carries out a series of binary classifiers to each pair of classes. In the one-against-all methods involves a parallel architecture made of N SVMs, one for each class. Each SVM solves a binary problem defined by a single information class against all others. The Binary Hierarchical Tree (BHT)-based methodologies implement the data processing using a hierarchical tree. This class of algorithms allows achieving better results in term of accuracy in a most time-effective way. BHT methods can be divided into BHT-Balanced Branches Strategy (BHT-BB) or BHT-One Against All Strategy (BHT-OAA). For the BHT-BB, the tree is defined in such a way that each node (SVM) discriminates between two groups of classes with similar cumulative prior probabilities. For BHT-OAA, the tree is defined in such a way that each node distinguishes between two groups of classes Ω_A and Ω_B. Ω_B represents the information class with the highest prior probability among those belonging to the union set (Ω_A U Ω_B) [262].

In the SVMs model, points represent the samples divided into classes by a construct an optimal hyperplane or a set of hyperplanes. The best hyperplane can separate data points of classes, and it is usually the plane that has the most significant margin between them [40]. There are two formulations of the SVM algorithm according to the possibility to linearly separable the classes or not.

1. Linearly Separable Case
   The equation (10) describes the discriminant function:

$$\mathrm{f(x)} = \mathbf{w} \cdot \mathbf{x} + \mathrm{b} \tag{10}$$

where:
x: the dimensional feature space;
w is orthogonal *f(x)*;
b is the bias.

These parameters have to be estimated based on the condition (11).

$$\mathrm{y}_i\,(\mathbf{w} \cdot \mathbf{x} + \mathrm{b}) > 0 \qquad \text{with i} = 1{,}2 \ldots \mathrm{N} \tag{11}$$

where:
y_i: class labels associated to **x**

The optimal hyperplane can maximize the distance between the closest training sample and the separating hyperplane (Figure 33). The training can be accomplished by solving this quadratic optimization problem (12):

$$\begin{cases} \text{minimize: } \frac{1}{2}\left|\left|w\right|\right|^2 \\ \text{subject to: } \mathrm{y}_i\ (\mathbf{w}\ \cdot \mathbf{x}\ + \mathrm{b})\ \geq 1 \quad \text{with i} = 1{,}2 \ldots\ \mathrm{N} \end{cases} \tag{12}$$

For real data, the separable case of classes is infrequent; thus, it is possible to have a linearly non-separable situation. To handle non-separable data, the SVM concept can be generalized combining to two criteria: margin maximization (as in the real case) and error minimization (to penalize the wrongly classified samples) (13).

$$\Psi(\mathbf{w}, \xi) = \frac{1}{2}\left|\left|w\right|\right|^2 + \mathrm{C}\sum_{i=1}^{N} \xi_i \tag{13}$$

where:
ξ_i : slack variables introduced to account for the non-separability of data;
C: regularization parameter that allows to control the penalty assigned to errors. C rules the trade-off between training error and generalization capability.

The (13) is subject to the following constraints:

$$\mathrm{y}_i\ (\mathbf{w}\ \cdot \mathbf{x}\ + \mathrm{b}) \geq\ 1 - \xi_i \qquad \text{with i} = 1{,}2 \ldots\ \mathrm{N}$$
$$\xi_i \geq 0 \qquad with\ i = 1{,}2, \ldots .N$$

2. Non-linear case (Kernel method)

The SVM can be improved in the non-separable case using a non-linear discriminant function (Figure 33). This modification of the SVM is called kernel SVM. The Kernel implementation of SVMs involves the problem definition of the kernel function, the kernel parameters and the regularization parameter C. To deal with this problem, recently automatic techniques based on gradient descent search have been developed to optimizing that parameter.

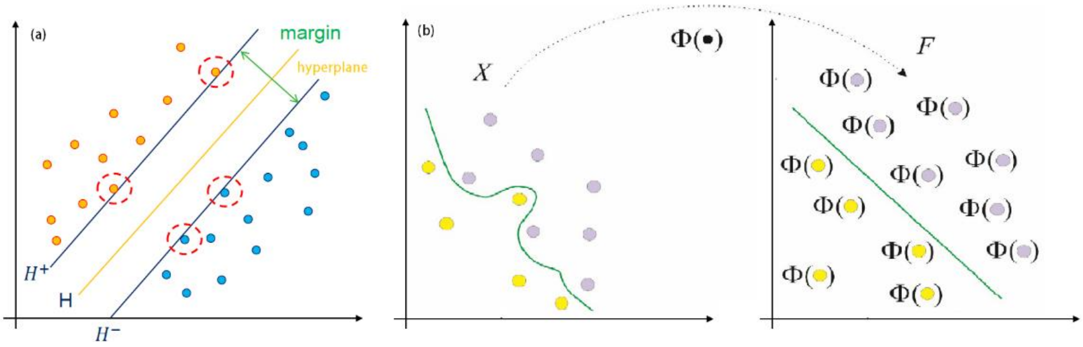


Figure 33. Optimal separating hyperplane in SVMs: (a) linear separable case and (b) non-linearly case.

Random Forest algorithm, proposed by Breiman [48] in 2001, combines multi-decision trees that operate as an ensemble trained with a bagging mechanism [48]. The bagging mechanism samples N (2/3) random bootstraps of the training set with replacement. The remaining 1/3 samples are referred as out-of-bag (OOB) data, which are used for inner cross-validation to evaluate the classification accuracy [263]. The advantage of this procedure is higher generalization capability due to the use of a random selected subset of predictor variables and the less correlation among trees. The possibility to measure the feature importance is another essential advantage of RF, because it allows understanding their contribution to the accuracy. The higher number of trees makes the algorithms more accurate than a simple decision tree [264]. Random Forest is based on the binary recursive partitioning trees using statistical variables (Figure 34).

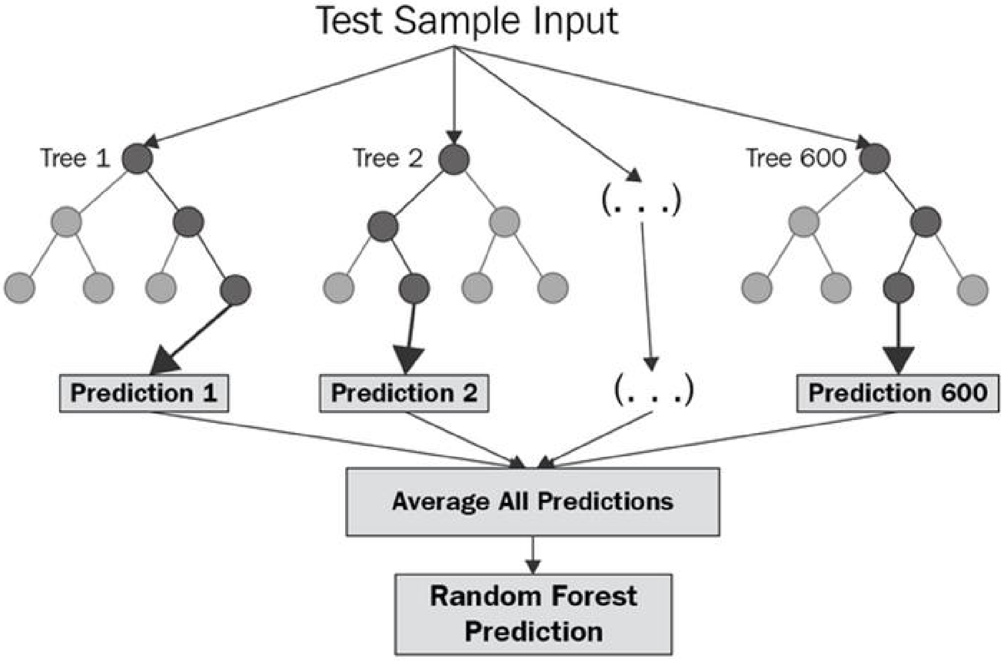


Figure 34. A general random forest conceptual schema.

Implementing Random forest requires, as in the case of decision tree (DT) algorithms, the choice of the attribute selection measures. The most popular attribute selection measures in decision tree are the Information Gain Ratio criterion and the Gini Index [48]. The information Gain Ratio is based on the measure of the variation of entropy (*H*) according to the (14):

$$\varDelta H = H - \frac{m_L}{m} H_L - \frac{m_R}{m} H_R \tag{14}$$

where:

H is the entropy defined as $-\sum p_k \, log_2 \, (p_k)$ where pk is the proportion of instances belonging to class *k* (with *k* = 1,2, …*k*) *m* represents the total number of instances and *m_k* are the instances belonging to a class *K* (with *k* = 1,2, …*k*).

GINI index (Q) instead is defined as following (15):

$$Q = 1 - \sum_{i=1}^{k} p_i^2 \tag{15}$$

where:
k: number of classes;
p_i is the proportion of class *k* observations in the node.

The GINI index measures the *purity* of classification at a node. Large values of a GINI index represent an impure node. According to the splitting criteria, a candidate split creates two descendant nodes and the splitting is chosen to minimize the following (16):

$$Q_{split} = n_L Q_L + n_R Q_R \tag{16}$$

where:
$_{QL}$ and *$Q_{R:}$* the two descendants;
n_L and *$n_{R:}$* the sample size.

The trees grow without pruning until the terminal node. The GINI index is less computationally intensive than the information Gain, thus is the most popular splitting criteria.

To characterize the performance of semantic segmentation, either for the RF and the SVM, the accuracy assessment can be achieved with different parameters based on the error matrix. According to the literature, the selected parameters are the following [193], [258]:

- *overall accuracy* (OA) that is the ratio of the total number of correctly classified pixels with respect to the total number of reference pixels;
- *user's accuracy* (UA) that is the ratio of pixels correctly classified in given class with respect to all pixels classified in the specific category.

Moreover, for RF, *out of bag score* can be considered. The *out-of-bag score* (OOB) indicates the model generalization, and it is computed as the number of a correctly predicted sample from the out-of-bag samples [264].

4.2.2 Object detection

Despite the semantic segmentation, the goal of object detection is locating the class-specific or class-agnostic bounding boxes. Object detection allows, thus, to determine if an image (RGB, hyperspectral) contains one or more objects belonging to the class of interest and locate the position of each predicted object in the image. The term 'object' refers in remote sensing application to its generalized form, including man-made objects (e.g. vehicles, ships, buildings, etc.). Despite the semantic segmentation which is a well-established technique in remote sensing, the object detection is still a challenging topic because it suffers from large variations in object appearance caused by viewpoint and the scale variation, occlusion, complex background, illumination, shadow, but on the other hand offer big opportunities in the field of real-time and multiscale application. Based on the cited differences between these two interpretation tasks, object detection requires more complex algorithms to accomplish it purposes.

In recent years, with the emergence of a variety of deep learning models, especially Convolutional Neural Networks (CNNs) were introduced in remote sensing applications. It is possible to classify the object detection methods in two streams: region proposal- based methods (two-stage algorithms) and regression-based (one stage) methods. As reported in the review proposed by Li et al. [265], the regional proposal -based methods solve the problem of detection in two steps using a CNN for creating object candidates and a classifier for defining the class. Some examples are the Region Based Convolutional Neural Network (R-CNN), Spatial Pyramid Pooling Network (SPP-NET), Fast R-CNN [266], and Faster R-CNN [54]. The regression-based methods can perform object detection tasks with a single neural network structure converting the classification and localization steps of the object detection task into a regression problem. You Only Look Once (YOLO), and Single Shot MultiBox Detector (SSD) are some example.

The main difference between the Object-Based (OB) methods is related to the generation of region proposal. This difference implies a distinct level of speed and level of accuracy. The regression-based approaches are faster, while the two regional-based methods are more accurate.

This section introduces a detailed description of the Faster R-CNN, because among the algorithms cited above, it represents the trade-off between computational time and accuracy and it shows some promising results as multiscale and quasi-real-time algorithm for remote sensing applications [267].

Faster R-CNN framework is a regional-based framework and is composed of three components: the backbone convolutional network (e.g. ResNET, AlexNET), a Region Proposal Network (RPN), and Fast R-CNN detector [54]. Figure 35 shows a generic RPN-based architecture for generic object detection.

The backbone is a feature extraction network, pre-trained in standard practice. In the first stage, the RPN, a deep fully convolutional network, predicts object locations and scores at the same time. In the second stage, Fast R-CNN handles region detection.

The input of the backbone is an image with a maximum size of 600 pixels (height), and 1024 (the width). The backbone output features (H×W) are smaller than the original image depending on the backbone stride. In Liu et al., 2019 [35], the stride is 16 [268].

RPN has to identify the object and discard the background. Then, for each item, RPN has to learn the location and the estimated size. To achieve these goals, it uses the last layer of feature map extracted by the CNN backbone and for each location in the feature map initializes k reference boxes, called anchors. Anchors indicates possible objects in a defined location with different scales (area of the bounding box) and aspect ratio (*H/W*). The scale and the aspect ratio sets allow dealing with different shapes and scales of the detection window [268].

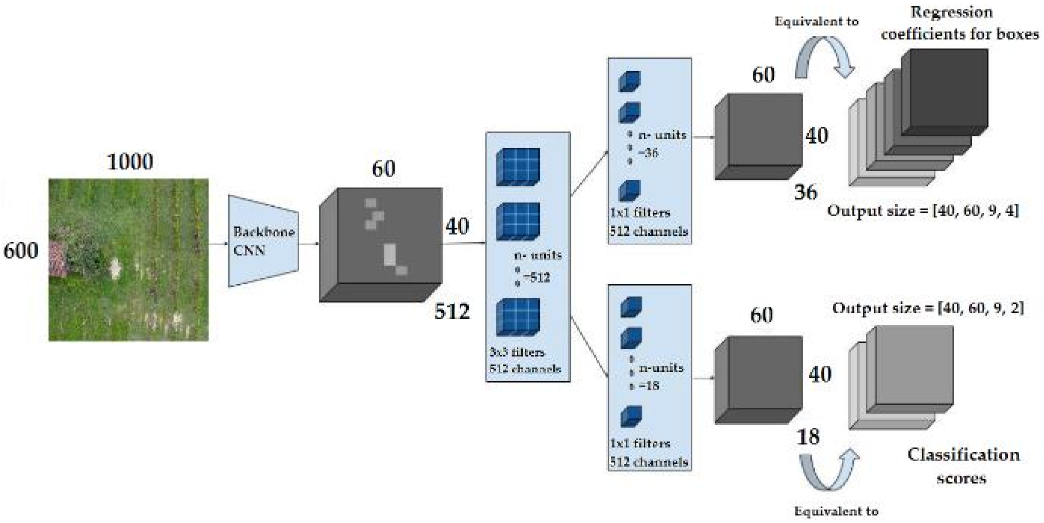


Figure 35. The Region Proposal Network (RPN) architecture [269].

The anchor box number *k* is defined considering the possible combinations of scales and aspect ratio (Figure 36). For a set of 3 scales and 3 aspect ratio, 9 anchors box is used.

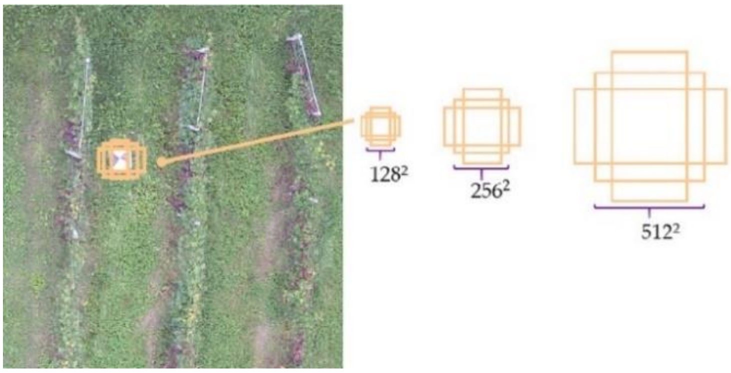


Figure 36. Example of anchors on real case images on the left. On the right, anchor scales (1282, 2562, 5122) and aspect ratios (1:1- squared shape, 1:2- horizontal rectangular shape and 2:1- vertical rectangular shape) for the PASCAL challenge [269].

Each anchor is mapped with an objectness score to a lower-dimensional vector. The objectness score indicates the membership to a set of object classes (s_{obj}) versus background (s_b). The positive score is assigned according to two different conditions: (1) the highest Intersection-over-Union (*IoU*) overlap with a ground-truth or (2) an IoU overlap higher than a threshold (in literature it is set to 0.7) with any ground-truth. The negative label is assigned to a non-positive anchor with an IoU less than a threshold (0.3). Moreover, it is possible that for a single ground-truth box, positive labels are assigned to multiple anchors. Thus, a Non-Maximum Suppression algorithm (NMS) reduces the redundancy of the anchor. It uses the IoU between each proposal and the most likely proposal. The IoU values have to be greater than a threshold (0.7) to select the ROIs with the highest probability to contain an object.

After defined the object proposal, a 3×3 convolutional layer with 512 units is applied to return a 512-d feature map for every location. The output of this last step is fed into two sibling fully connected layers which are 1×1 convolution layer with 18 units for object classification and 1×1 convolution with 36 units for bounding box regression. The classification branch gives an output of size (H×W×18) and indicates, for each feature map point, the probability of containing an object within all k anchor boxes (confidence score). The regression branch gives and output of size (H×W×36) and indicates bounding box coordinates.

The Faster-RCNN, as also defined for the Fast RCNN [266], loss function combines the losses of classification and bounding box regression as described in (17):

$$L(\{pi\},\{ti\}) = \frac{1}{N_{cls}}\sum_i L_{cls}(p_i, p_i^*) + \lambda\frac{1}{N_{box}}\sum_i p_i^* \; L_{reg}(t_i, t_i^*)$$

(17)

where:

i: index of an anchor;

N_{cls}, L_{reg} and λ: the normalization terms and the weight respectively;

p_i : predicted probability of anchor I being an object;

p_i^*: ground truth object label (1 for an object, 0 for not object);

t_i: vector of 4 parametrized coordinates of the predicted box;

t_i^*: ground truth box coordinate.

L_{cls}, as defined in (18), is the log loss function over two classes that are object and background in a binary case.

$$L_{cls}(p_i, p_i^*) = -p_i^* log, p_i - (1 - p_i^*)(1 - log, p_i) \quad (18)$$

where:

p_i: predicted probability of anchor I being an object;

p_i^*: ground truth object label (1 for an object, 0 for not object).

L_{reg} is defined as (19). The regression loss is activated only for positive anchors.

$$L_{reg}(t_i, t_i^*) = R\,(t_i - t_i^*) \tag{19}$$

where:
t_i: vector of 4 parametrized coordinates of the predicted box
ti^* : ground truth box coordinate
R: robust loss function (smooth L_1)

Average precision (AP), recall, and Intersection-Over-Union are used as evaluation metrics for Faster R-CNN. The Average Precision is the mean precision at a set of eleven equally conventional spaced recall levels [0,0.1,...,1] (20) [270], [271] :

$$\text{AP} = \frac{1}{11} \sum_{r \in \{0,0.1,\ldots,1\}} p_{interp}\,(r) \tag{20}$$

where:
p: the precision at each recall level r, interpolated by taking the maximum precision measured as in (21):

$$p_{inter} = \max p(r) \tag{21}$$

where:
p(r) is the measured precision at recall r.

The case in which the bounding box sufficiently overlaps the ground truth is defined as true positives (TP). False Positive (FP) is the case in which the bounding box overlaps with the ground truth insufficiently. False negatives (FN) are the ground-truth that could not be detected.

The *IoU* determines whether the proposed bounding box overlaps with the ground truth sufficiently, and it is used to measure the accuracy of object detection. It is defined as (22) [268]:

$$\text{IOU} = \frac{area\,(b \cap b^g)}{area\,(b \cup b^g)} \tag{22}$$

where:
b: predicted bounding box;
b^g : ground-truth
The IOU must be greater than a fixed threshold, typically set at 0.5 (50%).

Chapter 5

Semantic segmentation using hyperspectral and multispectral images

UAV hyperspectral and multispectral data are very high spatial and spectral resolution data, as explained in Chapter 2. Specifically, the hyperspectral data is more often composed by more than one hundred bands are difficult to manage in real- time processing because of their dimensions. Thus, a near-real-time automatic information extraction process, in the case of hyperspectral data, have to deal with [241]: the high data dimensionality that can affect not only the acquisition time, the storage and the transmission, but also the interpretation performances, and the spectral redundancy, i.e. the substantial overlap of information content over the bands per pixel.

The main objective of this chapter is selecting the most appropriate efficient, timely and cost-effective approach for the interpretation of hyperspectral and multispectral data. To test the semantic segmentation on hyperspectral and multispectral data, the specific chosen application as mentioned in Chapter 1 is the detection of the ice for monitoring and inspection of aircraft in de-icing operation.

The investigation has been developed within the activities of the European project, called Spectral Evidence of the Ice (SEI), and to understand the context, section 5.1 provides a brief description of the project. Moreover, section 5.2 define the main concept of the ice accumulation process. To perform the experimental analysis, a procedure for the generation of ice sample is presented in section 5.3. For ice detection, a semantic segmentation approach using two machine learning algorithms is tested. In details, the hyperparameter tuning of Random Forest and Support Vector Machine is performed for the specific application (section 5.5). Furthermore, facing the problems of high dimensionality and spectral redundancy, a methodology for band reduction is presented (section 5.4). To highlight the effect of the dimensionality reduction on the automatic information extraction method in term of accuracy and processing time, using different types of data, i.e. hyperspectral data, reduced hyperspectral data and multispectral data, a comparative analysis among algorithms, RF and SVM, applied with these data is shown (section 5.6). The analysis concerning the band reduction and the classification algorithms can be used to select the most suitable sensors for the proposed application and the best algorithm in terms of accuracy and processing time.

5.1 MANUNET SEI project: Spectral Evidence of Ice (SEI) for aircraft de-icing operations

As mentioned above, the experimental analysis has been carried out within the SEI project activities. The project research has been co-financed from European Union and both Greek and Italian national funds. The Greek co-financing framework is the Action “Business Support for Research Projects” through the Operational Program WEST GREECE 2014-2020 (project code: ΔΕΔΕ8-0028543, MNET18/ICT-3438). Instead, for the Italian parties, has been co-financed by the Piedmont Region funds within the framework of the Action “MANUNET III - POR FESR 2014-2020 (reference call: MANUNET III 2018, project code: MNET18/ICT-3438)”. Thus, the SEI project has a consortium of five partners: two Greek partners, i.e. IRIDA Labs S.A., and the University of Patras, Laboratory for Manufacturing Systems and Automation (LMS), and thee Italian partners, WPWEB SRL, Kite S.r.l., the Politecnico di Torino, PIC4SeR.

The SEI (Spectral Evidence of Ice) project [272], [273] proposes the design of an integrated, near real-time and automatic methodology and technological solution for aviation-related problems related to de-icing operations. The project pursues two different gaols: minimizing the risk to human safety [274] and reducing environmental damages. The hazard for flight safety is related to the ice formation that leads the aircraft balance in a loss of control, thus de-icing and anti-icing are necessary treatments during the winter [275]. On the other hand, cleaning process involves the employment of chemicals such as Ethylene Glycol (EG) or Propylene Glycol (PG) that can cause damage to the environment, in particular for the nearby surface and groundwater [276].

In literature, several devices have been presented as a solution for on-ground and in-flight ice inspection [277]–[279]. Ultrasonic, magnetostrictive, and electromagnetic sensors [280], mid-infrared sensor [281] have been designed. Even if the spectral sensors can be considered emerging technology in this application due to their high resolution, they are not yet appropriately developed [282].

Moreover, for information retrieval, the definition of devices for data collection are not enough. Automatics techniques for information extraction are needed. Classifiers such as Random Forest [14] and Support Vector Machine [15] (section 4.2.1) have been already used for material detection [283]. These classifiers, therefore, can handle high spectral dimensionality and noisy dataset [284], [285]. The physical characteristics of the ice and the variable atmospheric conditions make challenging the design of computer-oriented techniques for ice detection. Indeed, due to the physical properties of ice, the visual-based inspections are currently carried out for rime ice, snow and tactile-based for frost and clear ice by the trained and qualified ground crew or flight crew [286]. It is worth to underlines, however, that as far as the author is aware, the detection of the ice is a new application in UAV remote sensing.

To fulfil the gaps, a combined hardware-software solution is under development in the SEI project. The system has to handle four main operations: (1) automatic pre-flight inspection for the request of de-icing operation, (2) aircraft scanning to identify its position and shape, (3) ice detection to localize, quantify and define the type of ice, and (4) cleaning verification procedure with the feedback to the operator to visualize the ice-contaminated area and verify that the procedure is successfully done.

The hardware core technology is composed of a multi-sensor UAV platform equipped with a spectral sensor and the augmented reality (AR) devices. Figure 37 shows the physical schema of the whole system.

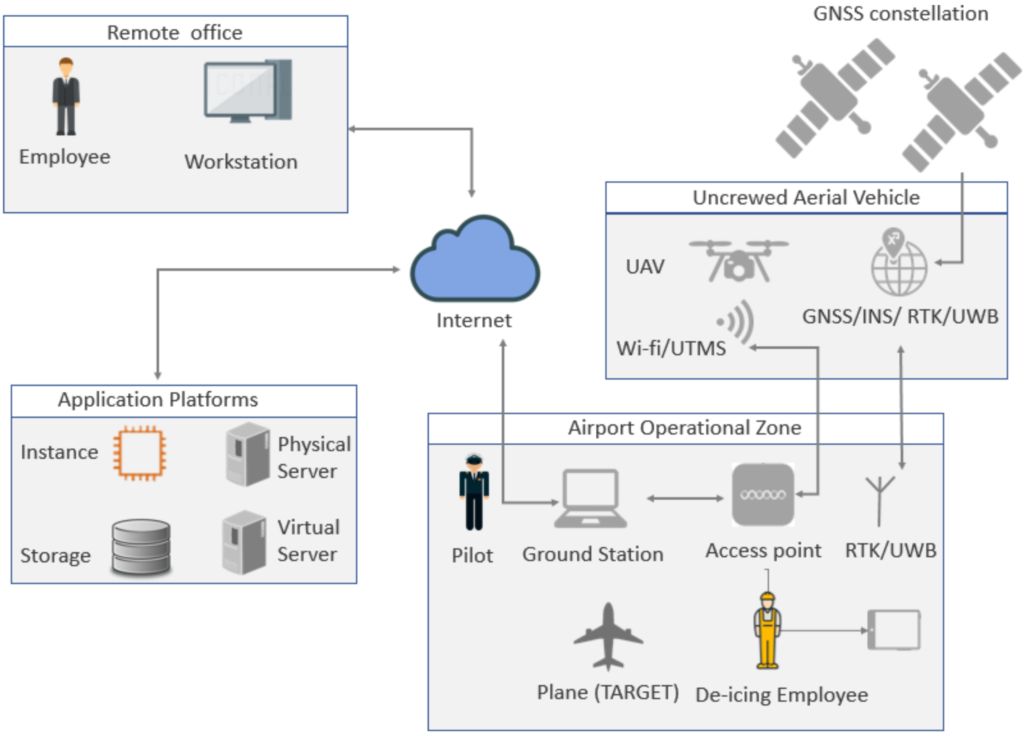


Figure 37. SEI: physical schema of the system.

The UAV task is the survey of the ice-contaminated area on the aircraft. The AR device helps the expert crew to visualize the results of the procedure. The software, elaborating the information collected by the drones, instead, identifies the location and the extension of the ice-contaminated area on the aircraft using an automatic methodology for geometric and radiometric detection of the ice. According to the goals of the project, the PIC4SeR contribution concerns to:

(1) the definition of the core technologies (imaging sensor, remote control and transmission module) for the ice detection using UAV, focusing on the imaging module;
(2) the identification of near-real-time procedure for hyperspectral data on-board management, i.e. band selection, pre-processing and Wi-Fi transmission;
(3) the selection of the near- real-time information extraction algorithm using hyperspectral or even multispectral data for the ice detection;
(4) identification of a procedure for hyperspectral data and navigation data synchronization (position and attitude) to correctly localize the ice on the aeroplane surface.

In this framework, this chapter is focused on the point (3), analysing algorithms for ice detection with a semantic segmentation approach and at the same time the effects of different types of data on the algorithm performances.

5.2 Ice accumulation process and definitions

To better understand the problem related to ice detection, in this section, the main concept on the ice accumulation process and the definition of the different ice types are presented.

Ice accumulation occurs when the supercooled droplets collide with a hard surface forming an ice film [287] with an air temperature between 0 and -20 °C [288]. As reported by the Federal Aviation Administration (FAA), the ice can be structural or in-flight ice and ground ice [289], [290]. The structural ice forms when the aircraft flights through visible water such as cloud droplets or rain. Ground ice, instead, may pile up when the aeroplane is parked due to precipitations. As mentioned in Manual of Aircraft Ground De-icing/Anti-icing Operations [275], the difference between in flight and on ground icing does not refer to ice features, but the ice's impacts on the flight safety, and the different de-icing procedures. Based on liquid water content, temperature, speed of the formation process, aircraft surface temperature and shape, particle concentration and size, three types of ice can be identified [291] (Figure 38): the rime ice, the clear ice and the mixed ice.

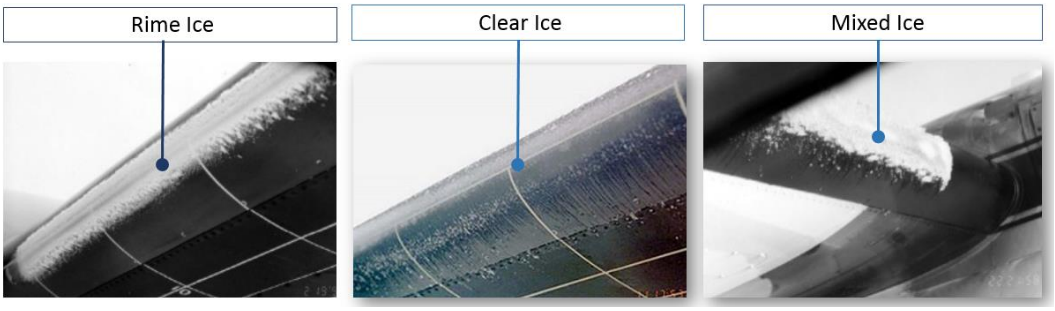


Figure 38. Ice types. (Photo credit: NASA, adapted from [291]).

The rime ice is a milky-white ice deposit. It usually forms on the leading edges, where anti-icing devices are installed [292]. It results from small droplets (with low water content) that rapidly freezing of at low temperature (<15 °C) after impacting with the aircraft surfaces. The rime density is lower than 200 – 300 kg/m^3, and it is composed mainly of discrete ice granules [293]. The clear or glaze ice is glassy transparent ice. It is the most hazardous ice type because it is the most difficult to detect through visual-based inspections and tends to cover the leading edge, greatly affecting airflow [292]. It is caused by large droplets that come backwards on the aircraft surface with slow freezing. The glaze ice density is higher than 800 or 900 kg/m^3 [294]. The mixed-ice has mixed characteristics of both rime and clear, and it occurs when mixed ices accumulate on the aircraft. Other examples of ice are frost, snow (or slush), fog, drizzle, rain (and their freezing states), and ice pellets. Frost occurs due to the deposited water vapour sublimation on the aircraft. The temperature, in this case, is lower or equal to the freezing. Precipitations of ice crystals induce snow or slush. Supercooled water droplets freezing upon the impact with the aircraft surface can cause fog, also known as freezing fog, and produces a coating of rime/clear ice. Indeed, uniform precipitations of liquid water particles based on the drop size and proximity can cause rain and drizzle. In details, the rain is characterized by well-separated drops with a diameter more than 0.5 mm; drizzle is formed by close drops with diameters less than 0.5 mm. Rain and drizzle, in the frozen state, can create ice deposits with a transparent appearance.

5.3 From ice sample generation to ice detection

The definition and experimental testing of the methodology for ice detection in a real case, i.e. at the airport, are very complex due to the following main reasons:

(1) the test site: the airport has severe constraints and security and safety regulations related the access, the use of UAV, the experimental tests involving operational spaces, i.e. parking area and hangar;

(2) the season: the winter is the optimal period to carry out the test; however, it has a greater logistical complexity due to the presence of ice and snow;

(3) the UAV: the realization of SEI drone prototype is a goal of the project, as mentioned in section 5.1. Moreover, it requires securing operation due to the integration of “heavy” sensors such as the hyperspectral camera and long trials and field tests which are still in progress; thus, the drone is not yet available for the experimental tests.

Therefore, a simulation was carried out to overcome the issue of not being able to perform acquisitions in the airport with the final UAV prototype in winter. A simulation dataset was acquired to demonstrate the feasibility of using hyper/multispectral sensors and testing the algorithms' performance. However, it is worth to underline that the method can be transferred to a UAV application.

Thus, the experimental analysis was carried out at Photogrammetry, Geomatics and GIS Laboratory of DIATI (Department of Environment, Land and Infrastructure Engineering) at Politecnico di Torino (Italy) [295]. First, ice samples were created in the laboratory. Then, a ground station was equipped with a hyperspectral camera (Senop Rikola) and a multispectral camera (MAPIR) for the data acquisition. The sensor’s specifications are presented in section 3.1.1. After that, radiometric calibration and ice detection with hyper/multispectral data were developed. Figure 39 presents the overall workflow.

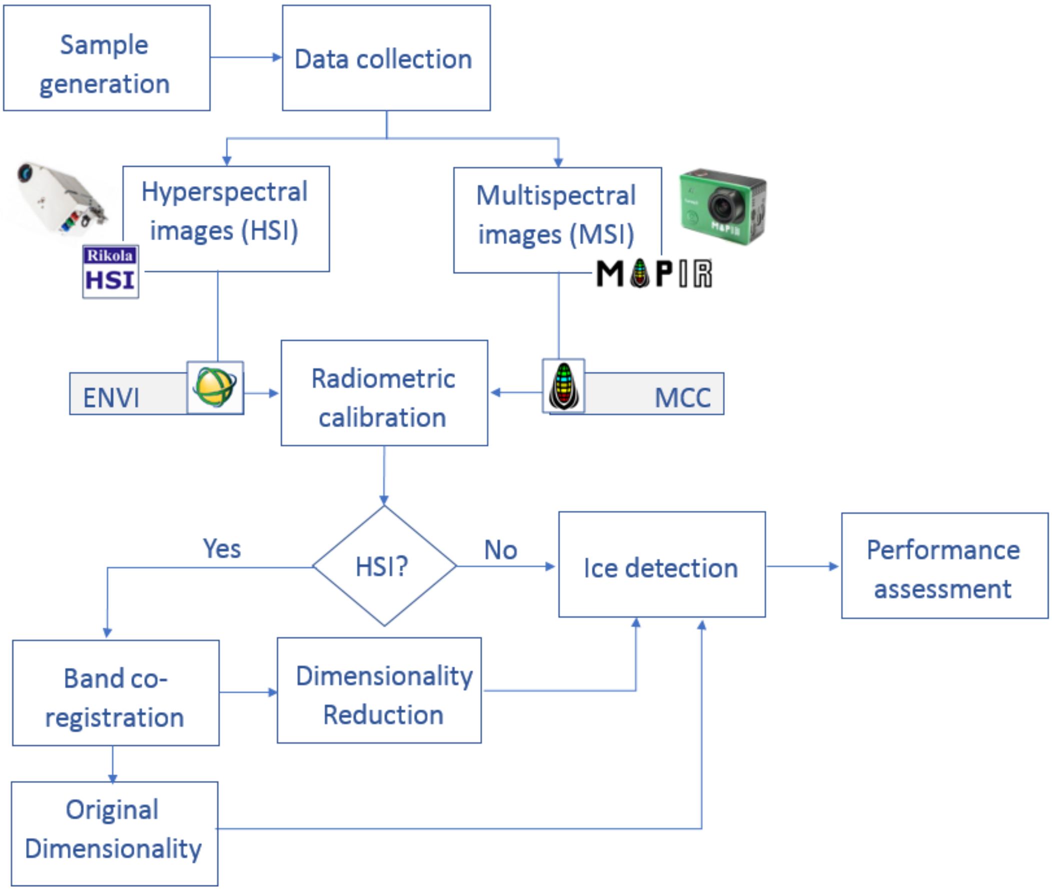


Figure 39. Methodology workflow.

In detail, for the sample generation, two types of ice, rime ice and clear ice, were taken into account. The rime ice and the clear ice were produced as much as similar to the real case. The rime ice was generated with a variable thickness (included between 2 and 6 cm) using the water vapour condensed at a temperature of -12° into the freezer. The clear ice was created by freezing tap water within plastic moulds at the same temperature. To reproduce ice block with different thickness (from 2 to 3 cm) different plastic moulds containing from 5 to 20 ml of water were chosen. Snow or other ice types mentioned in Section 5.2 are not included in these experiments because of our laboratory's lack of instruments for their production. Instead, for aircraft material, a section of the wing was retrieved. The aluminium panel was a section of a Socata MS.894 Rally Minerva with a size of 400 × 400 × 2 mm (Figure 40). As it is possible to notice from Figure 40, the clear ice on the freeze aluminium panel is difficult to detect. The panel was stored in the freezer to be a little bit iced as in the winter conditions. The ice blocks were placed on the section of an aircraft wing to simulate the typical conditions.

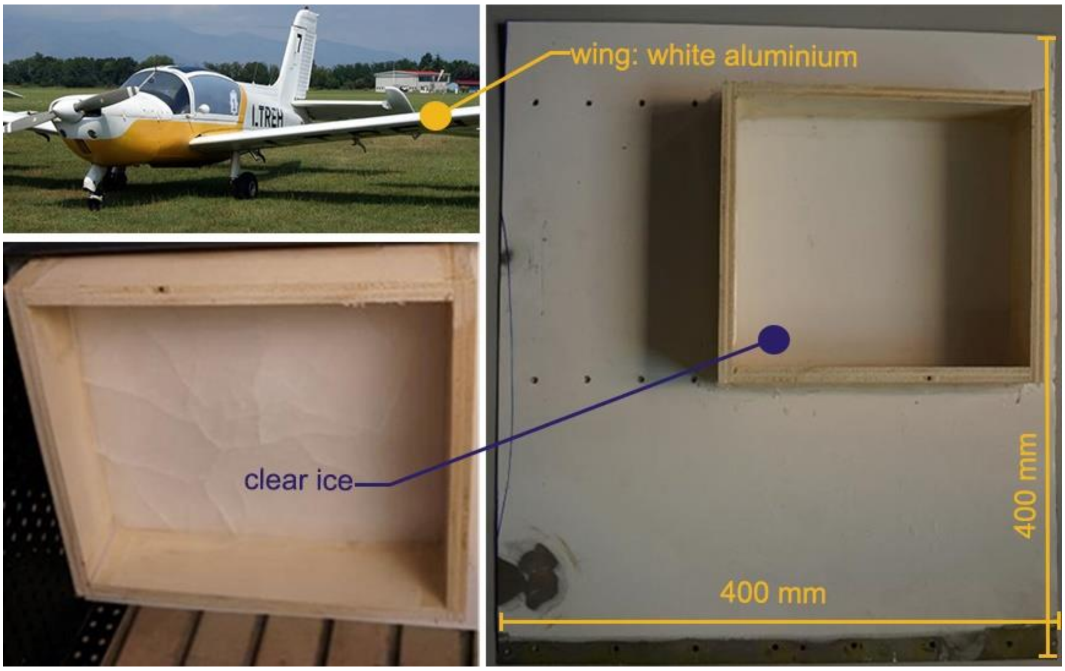


Figure 40. The wing section used for the experimental analysis and the clear ice.

Two measurement campaigns were performed in two different times, both in indoor and outdoor conditions. To simplify the identification of each campaign, an encoding *x_yyy_zzz* was defined, where: *x* identifies the environmental conditions, *I* for indoor or *O* for outdoor, *yyy* stands for the type of sensor, *HSI* or *MSI*, and *zzz* is the number of the bands. From these campaigns, the four datasets derive:

(1) I_HSI_100: composed of 10 hypercubes with 100 bands;

(2) I_MSI_3: includes 8 Red, Green, and Near-Infrared (RGN) images with 3 bands;

(3) O_HSI_100: encompasses 9 hypercubes with 100 bands;

(4) O_HSI_10: includes 9 hypercubes with 10 bands.

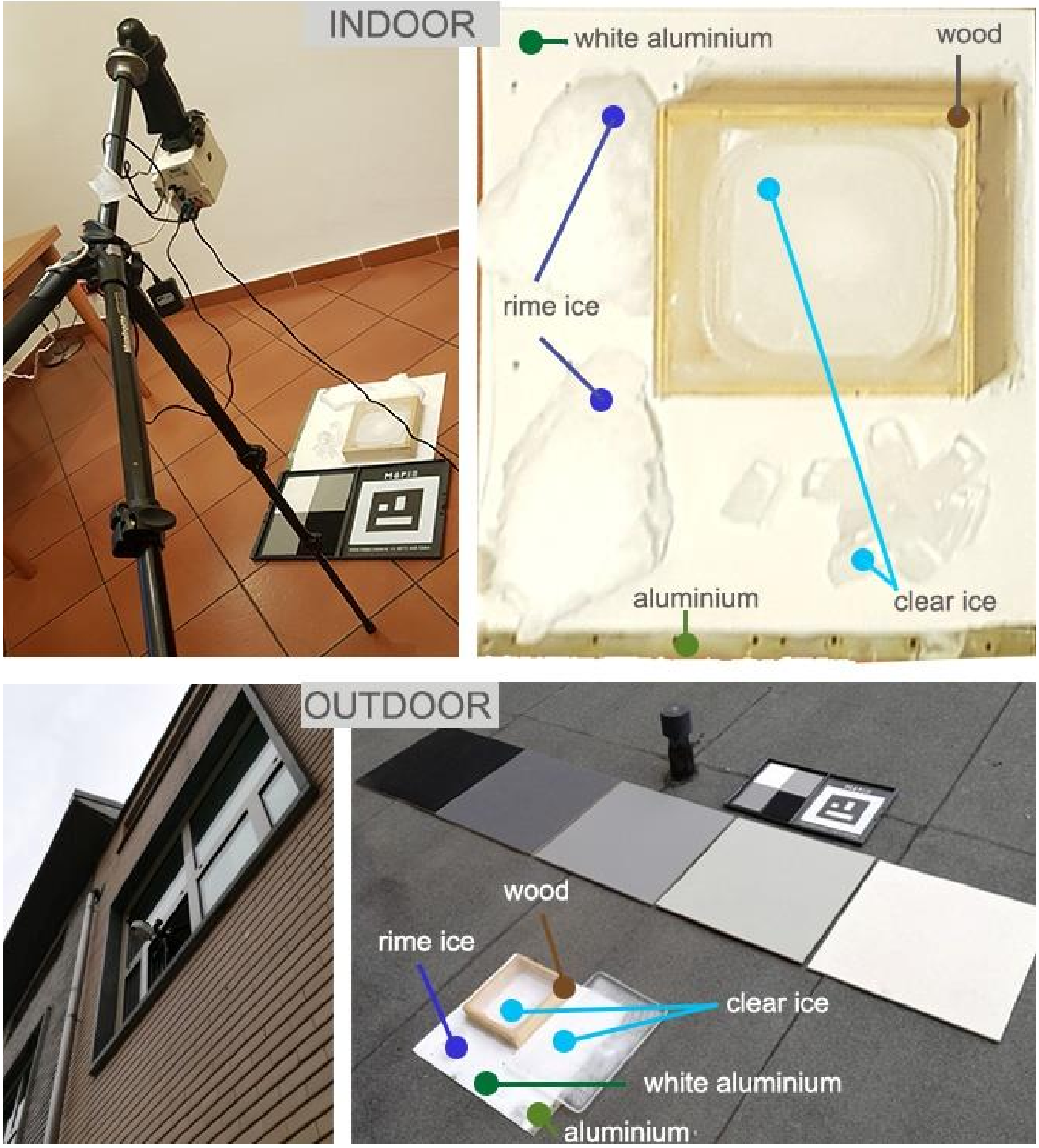


Figure 41. Indoor acquisition setup and material samples (above); outdoor acquisition setup and material samples (below).

Figure 41 presents the setup of acquisitions and material samples. The first two data collections (I_HSI_100 and I_MSI_3) were carried out simultaneously in an indoor environment, and the aluminium panel was located at 1.5 m of distance from the camera. A combination of lamps and natural light and several lamps were employed to simulate the various illumination conditions. The O_HSI_100 and O_HSI_10 acquisitions were performed outdoor, in distinct moments of the day, setting the camera at 5-meters distance from the objects. All data were collected, maintaining stable positions of the camera and varying slightly only the rotations.

The hyperspectral camera was used, in all the acquisitions, in manual mode connected to the computer through a USB cable. The image resolution was set equal to 1010 × 1010 pixels. According to the illumination condition, the integration time was fixed at 450 ms in I_HSI_100 case. Instead, in the O_HSI_100 and O_HSI_10, it was varying from 10 to 50 ms (with a step of 5 ms). The Rikola Hyperspectral Imager software v2.0 automatically created the sequence of the bands. The whole electromagnetic spectrum (from 502 to 906 nm), in the I_HSI_100 case, was covered to define the most significative bands and materials characteristics. The images were composed of 100 bands with a wavelength step of 4 nm and a Wide Full With Half Maximum resolution (FWHM). In the case (O_HSI_100) and (O_HSI_10), two different sizes of hypercubes have been considered: 100 bands and 10 bands, respectively. Since the third and the fourth campaigns were consequential to the analysis of the I_HSI_100, the band sequences were set based on results of section 5.4. For the I_MSI_3 dataset collection, ISO-800 as camera's sensitivity, and 1/15 s as the exposure time was set. The term "Test" refers, in this chapter, to each image with different environmental condition.

After the data collection, the hyperspectral images were co-registered based on the procedure and the MATLAB routine described in section 3.2.4. Then, all the HSI datasets and the MSI dataset were radiometrically calibrated. The HSI datasets were corrected by the Empirical Line Calibration tool of ENVI 4.7 [187]. In details, the calibration factor was computed using as references the MAPIR panel for the indoor dataset (Figure 42) and the EnsoMosaicMill targets for the outdoor dataset (Figure 43). Figure 44 and Figure 46 show the results of the co-registration and radiometric calibration process. The I_MSI_3 dataset calibration was performed using the MAPIR Camera Control (MCC) software package [188]. Figure 45 represents a comparison between an original multispectral image and the corrected version. Furthermore, for the multispectral case, after the radiometric calibration, an RGBN (Red, Green, Blue and Near-Infrared) image was generated using the *Composite Bands* tool of ArcGIS Pro 2.5.0 [296] (I_MSI_4). This step was important to understand the possible improvement of an additional band on the classification.

The use of HSI with 100 bands requires time both for the acquisition and classification processing. Therefore, a dimensionality reduction procedure was performed on the HSI datasets. The procedure, described in section 5.4, allows defining three new datasets.

Before proceeding with the ice detection, the identification of ground-truth had to be carried out. The identification of the ground-truth refers to the process of selection of regions-of-interest (ROIs). Each ROI that forms the spectral libraries contains a subset of samples.

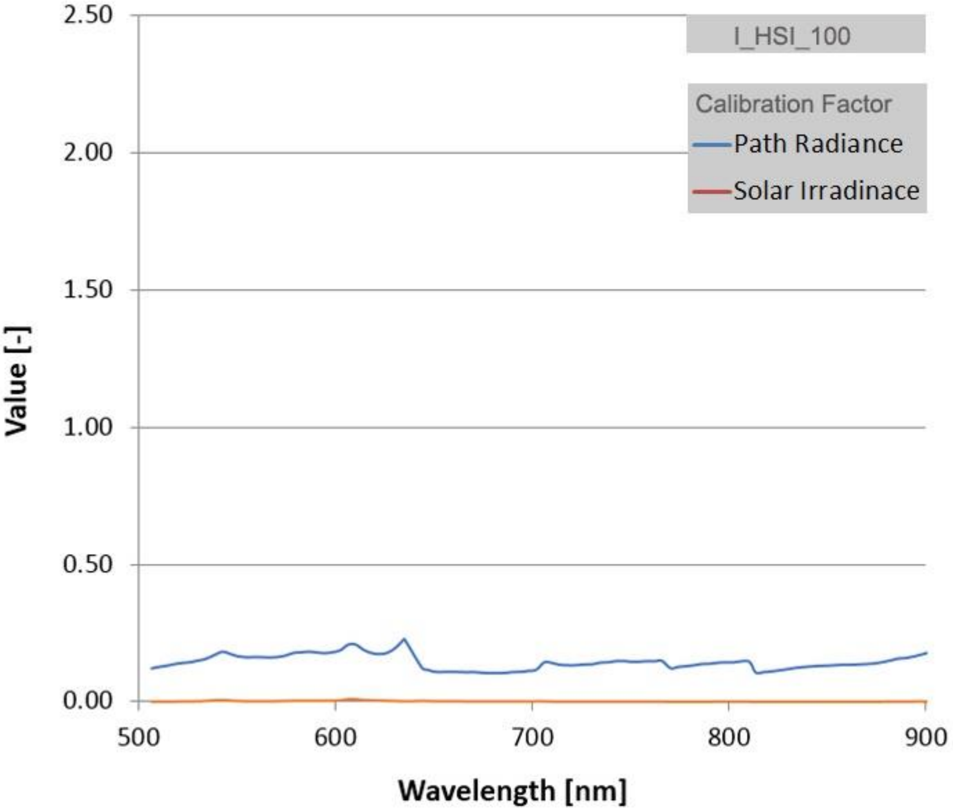


Figure 42. I_HSI_100: the calibration factor computed for the radiometric correction.

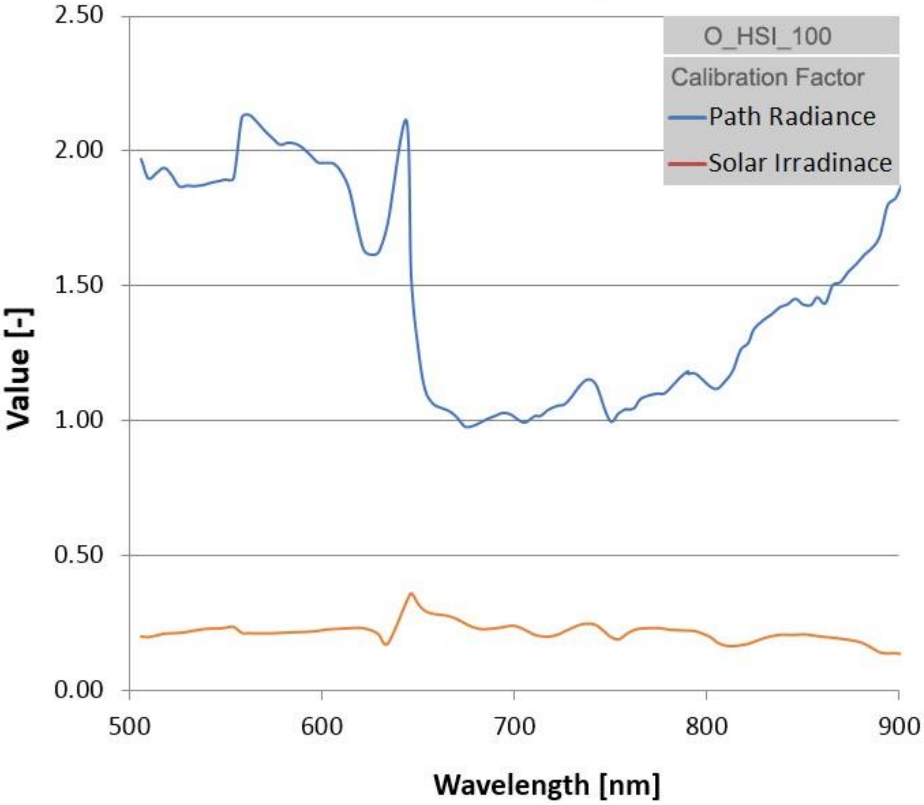


Figure 43. O_HSI_100: the calibration factor computed for the radiometric correction.

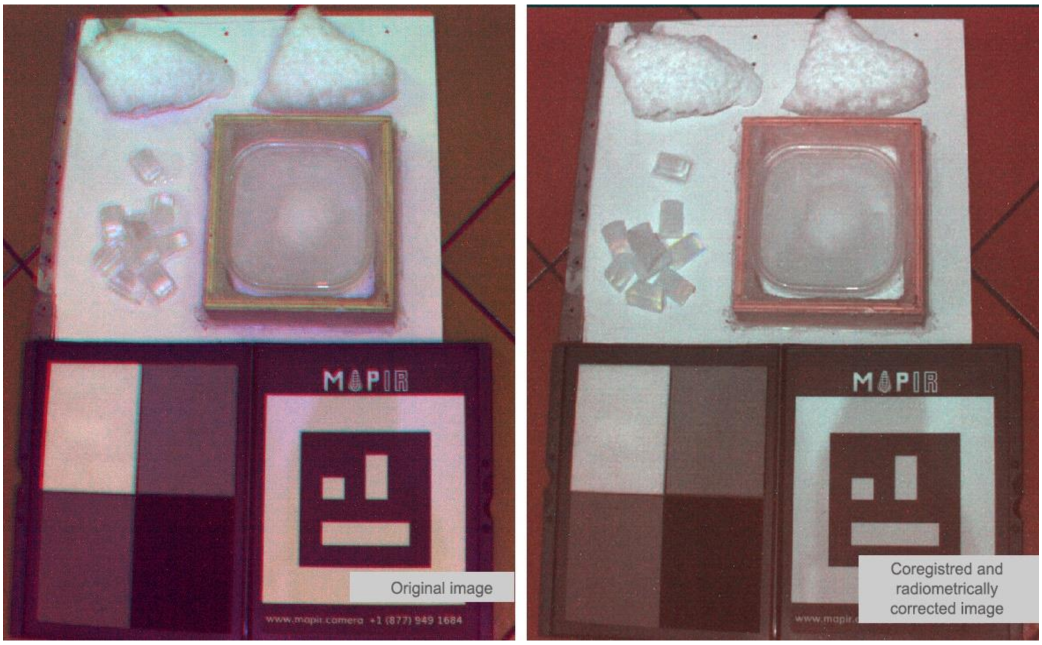

MAPIR
Original image
www.mapir.camera +1 (877) 949 1684
MAPIR
Coregistred and radiometrically corrected image

Figure 44. I_HSI_100: the comparison between an example of original HSI image (bands: 846.21, 643.12,517.86 nm) and its co-registered and radiometrically corrected version.

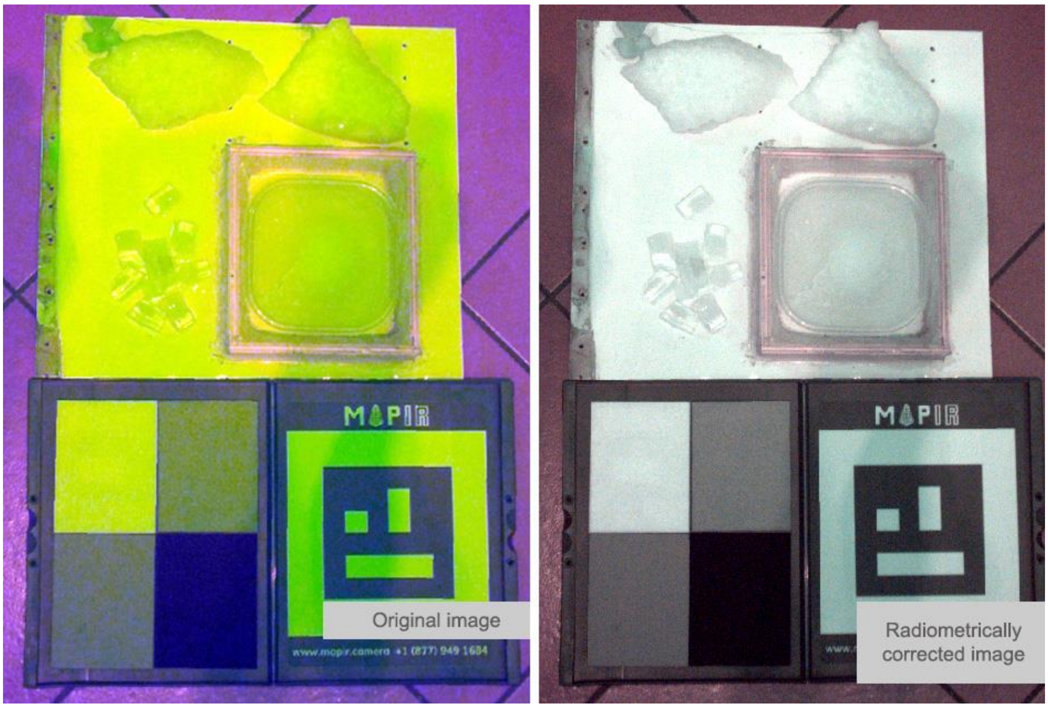

MAPIR
Original image
www.mapir.camera +1 (877) 949 1684
MAPIR
Radiometrically corrected image

Figure 45. I_MSI: the comparison between an example of original MSI image (RGN), and its corrected version.

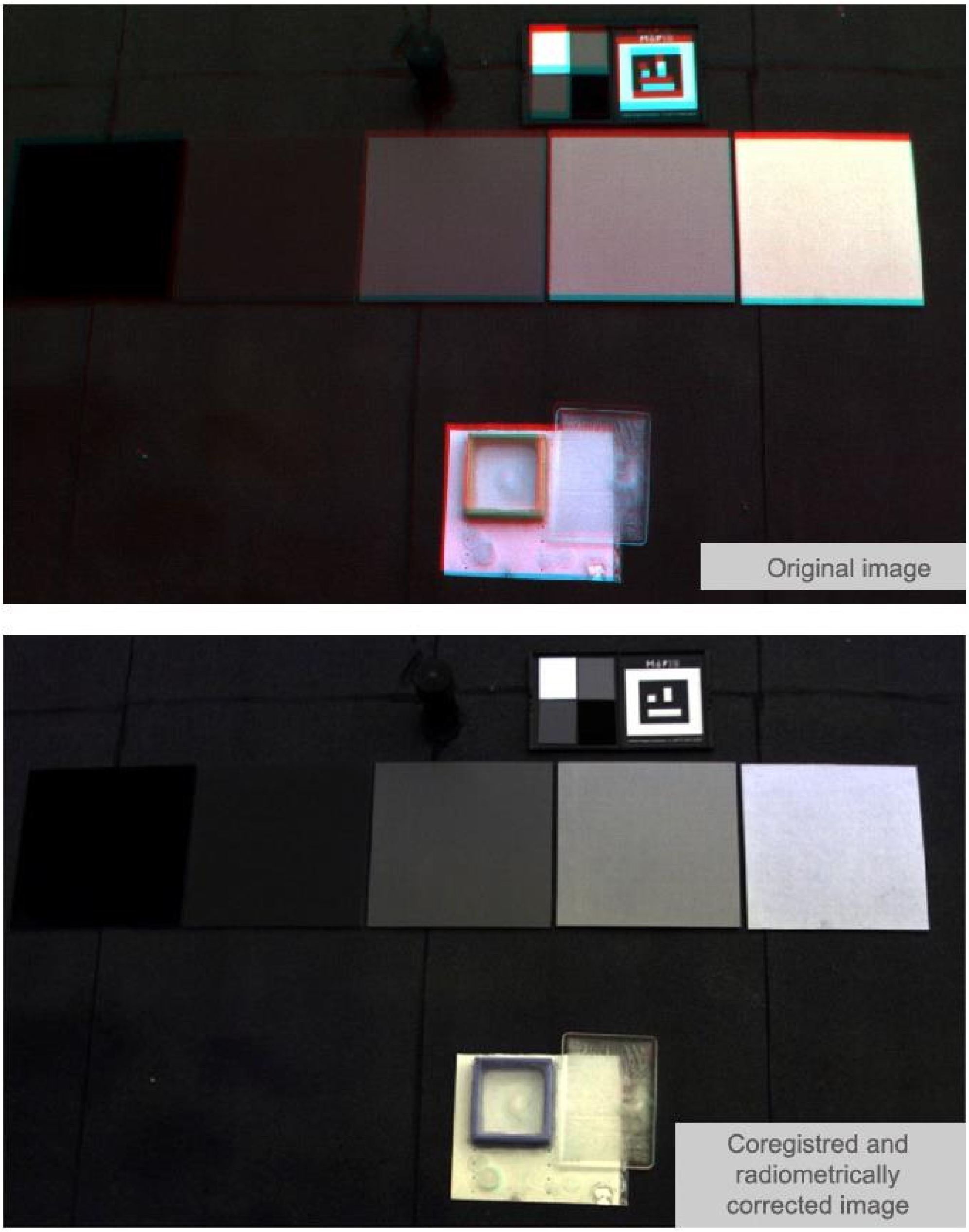


Figure 46. O_HSI_100: the comparison between an example of original HSI image (bands: 846.21, 643.12,517.86 nm), and its corrected version.

The sample subset belongs to the same class, and thus to the same material. ROIs must be defined only selecting pixels with pure spectra and excluding glint, shadows, and boundary pixels. Thus, some images were manually annotated. Figure 47 and Figure 48 represent the outputs of this step. Even if the representative classes were only the rime ice, the clear ice, and the white aluminium, a group of 10 classes were considered to characterize the background better and improve the performance of the classification. A single class for background materials could alter the accuracy of outcomes. The selected class are rime ice, clear ice, reference panel (white, black, grey 21%, and grey 27%, white aluminium, aluminium, floor tile, wood). The annotation process was performed by ArcGIS Pro 2.5.0 toolbox, that allows generating polygons as reference data for each class. The samples were collected based on visual interpretation. The number of samples per class varies based on image resolution of the HSI and MSI dataset. An average of 10,523 (hyperspectral) pixels, and 104,058 (multispectral) pixels per class have been annotated for the training and the validation of the semantic segmentation step.

Finally, after the data preparation, the hyperparameter tuning and the ice detection were performed according to the procedure reported in section 5.5 and section 5.6.

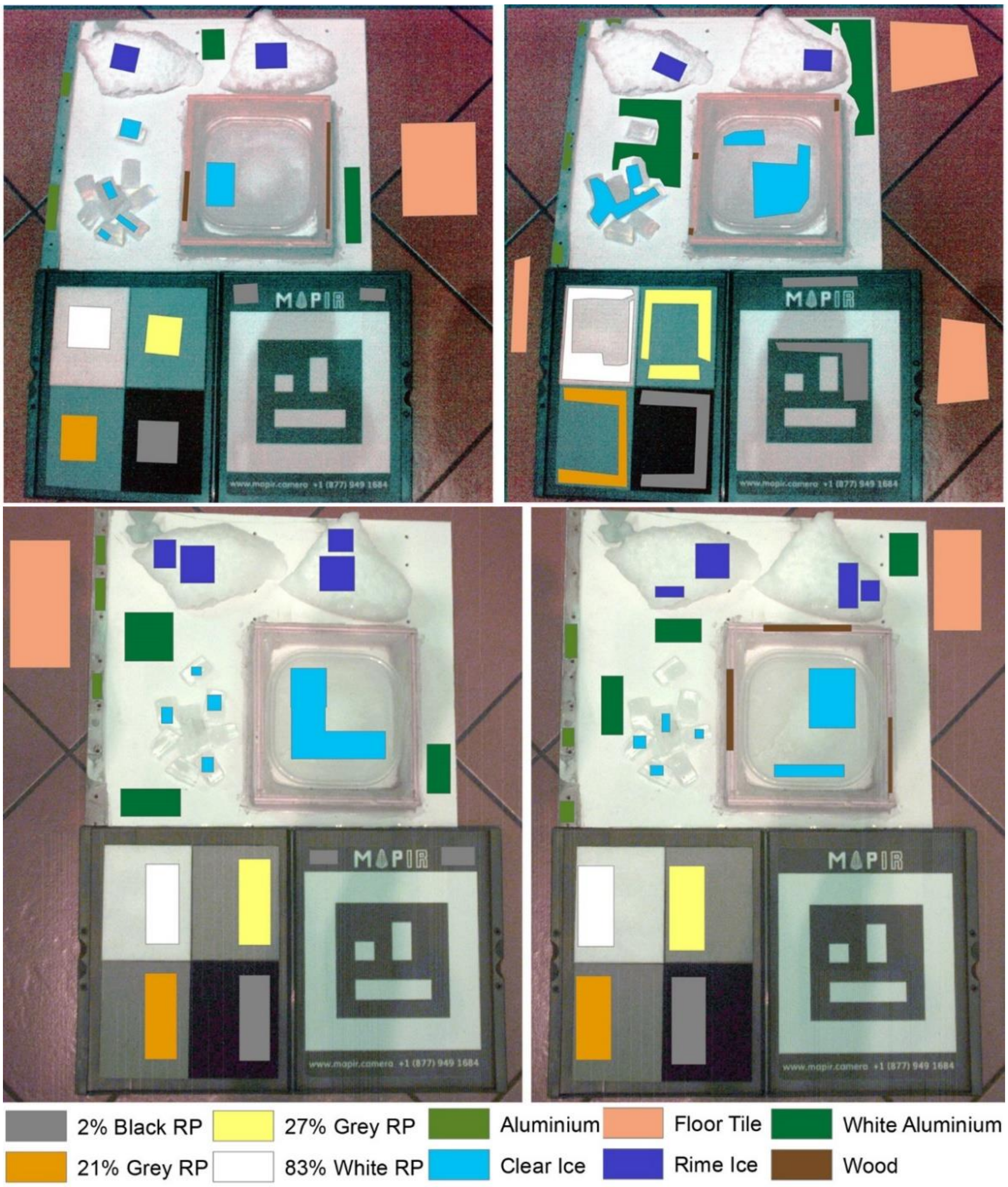


Figure 47. I_HSI_100: an example of an annotated image with reference data (bands: 846.21, 643.12,517.86 nm). Training set (on the left) and validation set (on the right). RP stands for Reference Panel (above). I_MSI: an example of an annotated image with reference data. Training set (on the left) and validation set (on the right). RP stands for Reference Panel (below).

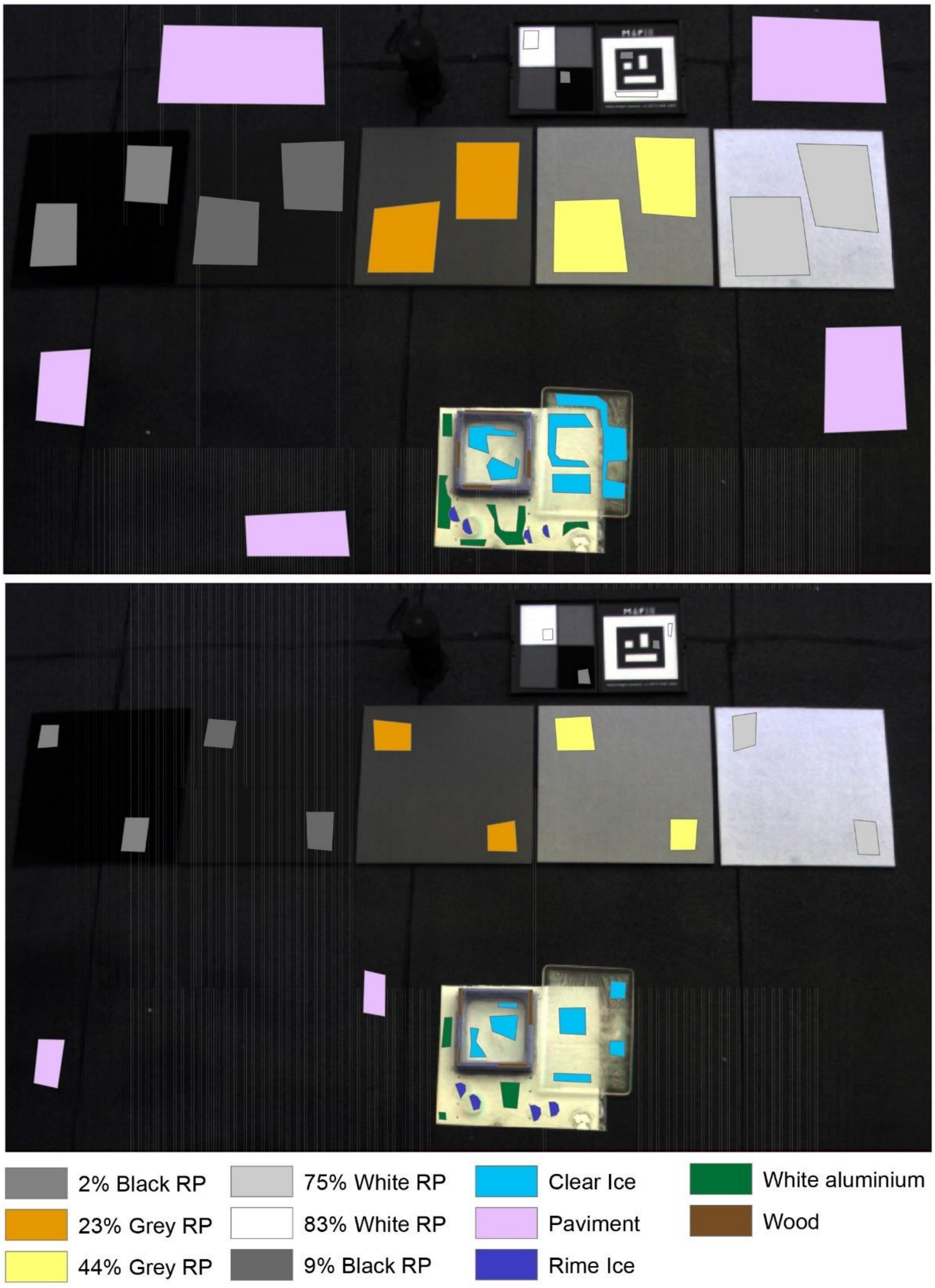


Figure 48. O_HSI_100: an example of an annotated image with reference data. Training set (above) and validation set (below). RP stands for Reference Panel.

5.4 Dimensionality Reduction of hyperspectral data

As mentioned in section 4.1, the Principal Component Analysis can be employed as a feature extraction and band selection technique to reduce spectral dimensionality. To understand the best approach for this application, both feature extraction and band selection methods were applied to both the hyperspectral datasets, I_HSI_100 and O_HSI_100. PCA was performed using the *Principal components tools* of ArcGIS Pro 2.5.0 [297].

First, the feature extraction was carried out to define the principal components (PCs). Then, the selected PCs were used for significative band selection.

In the feature extraction process, the number of PCs referred to the new dimensionality was defined. According to the criteria listed in section 4.1, the cumulative percentage of the total variation, the Kaiser's rule and the screen plot were evaluated. To better understand the feature extraction process, an example presents the results of one sample image for the I_HSI_100 dataset. Indeed, Table 14 and Figure 49 show that three PCs are the most representative, and the other can be discarded. As reported in Table 14, the first three PCs amount to the cumulative total variance equal to 90.31% (the value is included between 80% and 90%), and for the Kaiser's rule, the variance exceeds the 1%. Moreover, in the screen plot, the elbow is formed at the third component (Figure 49). It is possible to achieve the same results for the other images of the dataset.

Table 14. Principal component analysis on a I_HSI_100 sample image: eigenvalues and cumulative variance in percentage.

Component	Percent %	Cumulative%
1	80.0010	80.00
2	8.4818	88.48
3	1.8341	90.32
4	0.9935	91.31
5	0.5888	91.90

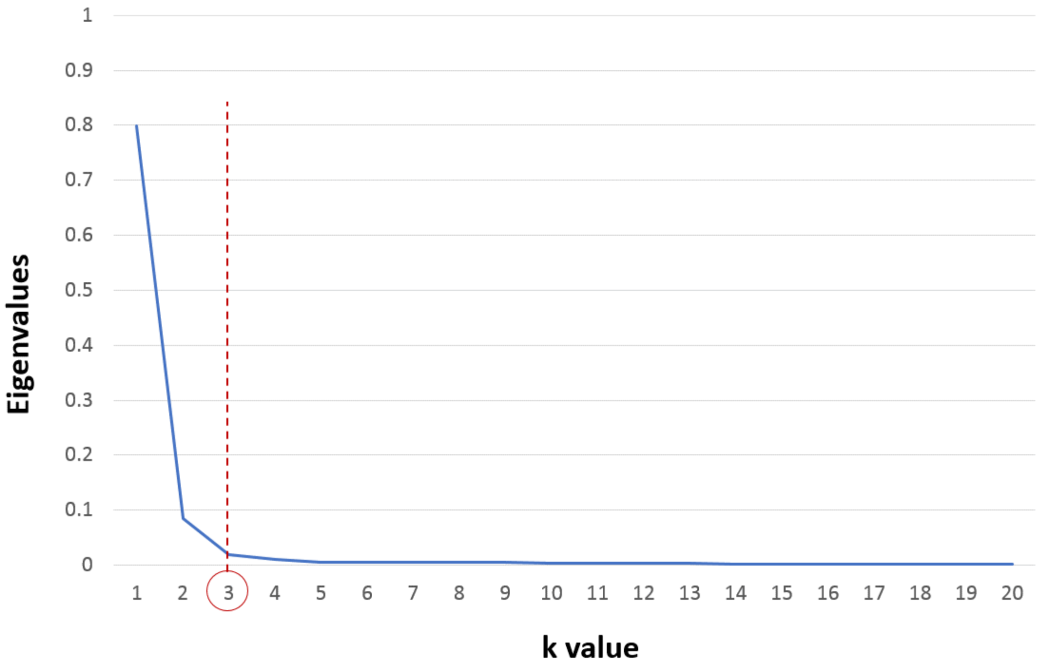


Figure 49. I_HSI_100: screen plot and zoom of the *elbow* for a sample image [298].

After the feature extraction, the chosen PCs were used for the band selection. In this case, original bands were identified through the eigenvectors of the principal components. The higher the significance of that band for the specific principal component, the higher the absolute value of the band eigenvector. Therefore, a threshold of eigenvector values for each component was defined to select the significative bands. The eigenvector plot shows the mutual relation between spikes of the function and the representative bands.

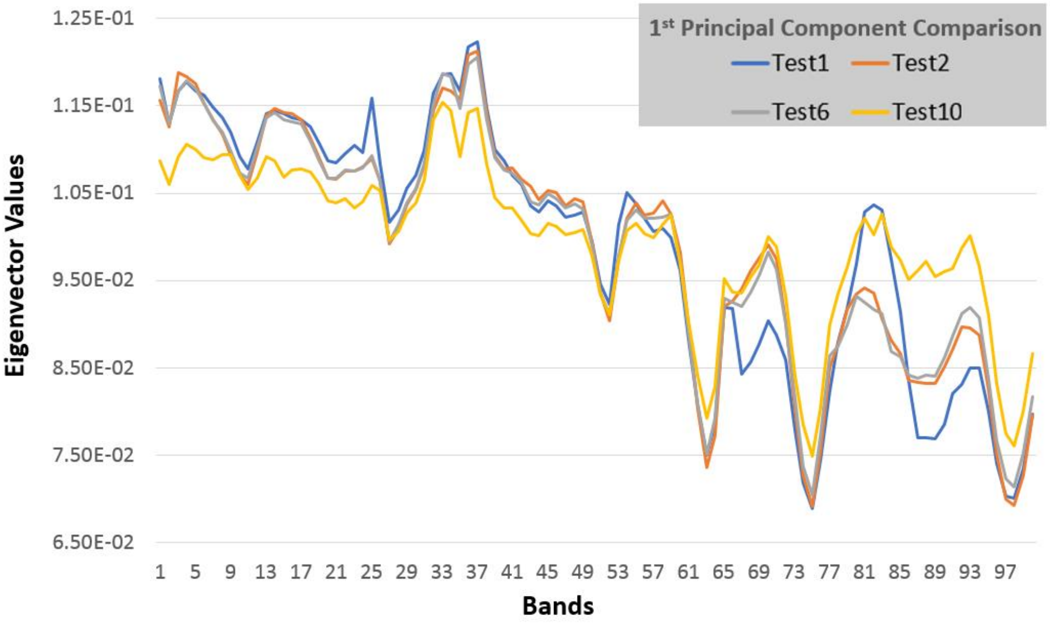


Figure 50. I_HSI_100: eigenvalues plot for the first principal component [298].

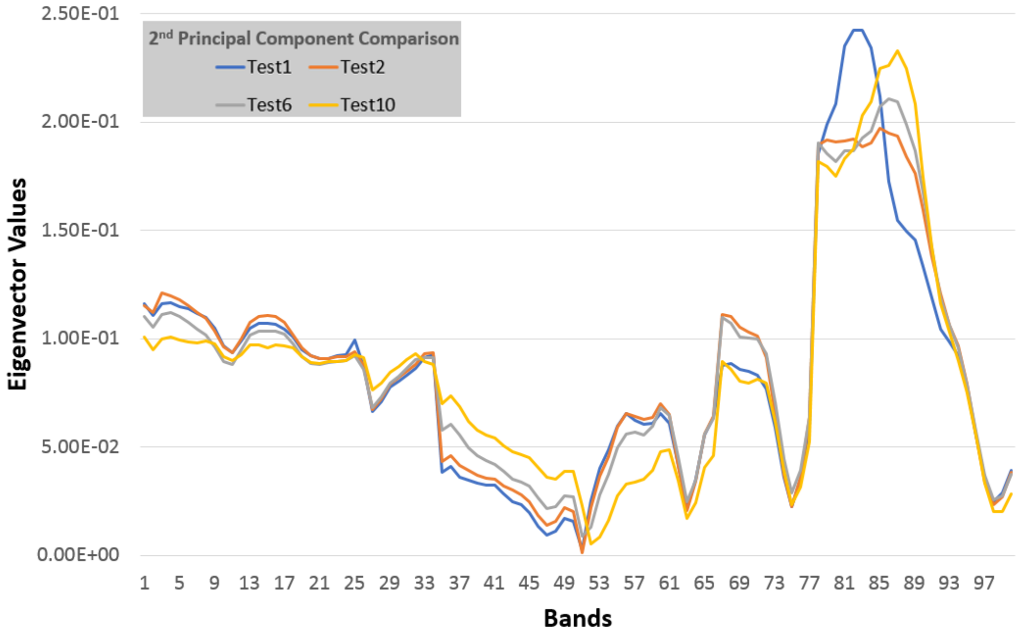


Figure 51. I_HSI_100: eigenvalues plot for the second principal component [298].

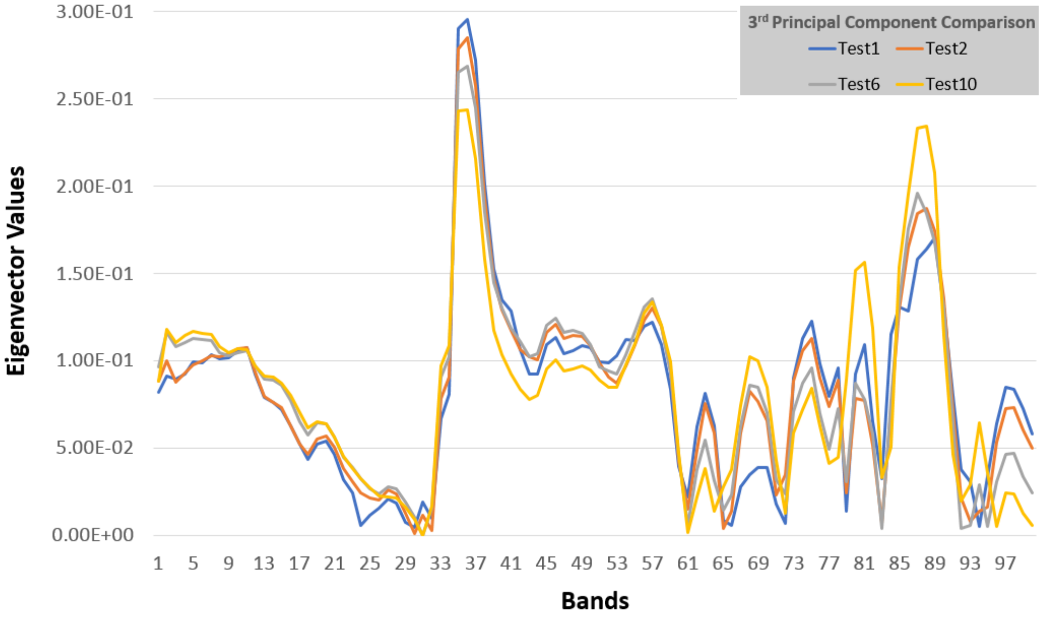


Figure 52. I_HSI_100: eigenvalues plot for the third principal component [298].

In details, the spikes in the eigenvectors function allow identifying the bands. Thus, Figure 50, Figure 51, and Figure 52 illustrate the plot (eigenvectors to the band number) of the three selected principal components in four representative images (Test1, Test2, Test6, Test10). These four images were characterized by different illuminations and state of ice to control the recurrence of the bands in several conditions.

Table 15 reports the significative bands extracted using the three PC of the four test images. It is worth to underline that each test presents some common bands. The dimensionality reduction, at this stage, is arbitrary, based on the application. Thus, it is possible to identify two other possible reductions: a new hypercube composed of 27 bands and one with 10 bands. In the first case, all the significative bands extracted from the four tests were picked up, which means includes the bands 1 (506.31 nm), 3–7 (from 514.48 to 530.11 nm), 14 (558.28 nm), 25 (602.47 nm), 32–38 (from 630.2 to 654.19 nm), 78–89 (from 817.58 to 861.65 nm). In the latter, only the common bands of each principal component are selected. Hypercubes can be composed of 10 bands included in the following bins: 4–5 (from 518.12 to 522.48 nm), 33–37 (from 634.36 to 650.38 nm), 83–85 (from 837.98 to 846.21 nm).

Table 15. I_HSI_100: significative bands in the representative test extracted from the three principal components.

Principal Component	Test1	Test2	Test6	Test10
1	1, 3–7, 25, 32–38	1, 3–6, 14, 33–37	1, 3–6, 32–37	4, 5, 32–34, 36, 37
2	80–85	80–85	78–89	83–89
3	35–37	35–37	35–37	35–37, 86–89

The outcomes of the dimensionality reduction are three new datasets. To simplify the identification of each dataset, an encoding *x_yyy_zz* was defined, where: *x* identifies the environmental conditions, *I* for indoor or *O* for outdoor, *yyy* stands for the hypercube type, *PC* for the hypercubes composed by the principal components and HSI for the reduced original bands, and *zz* is the number of the component or bands. Thus, the dataset are:

(1) I_PC_3: 10 new hypercubes composed by the three PCs;
(2) I_HSI_10: 10 new images with 10 bands;
(3) I_HSI_27: 10 hypercubes with 27 bands.

The I_PC_3 hypercubes were generated using the *Principal Component Analysis* toolbox of ArcGIS Pro 2.5.0. In other cases, the hypercubes were decomposed and composed respectively through a customized MATLAB routine (Appendix A) and the *Composite Bands* tool of ArcGIS Pro 2.5.0 [296].

The same procedure was applied to the O_HSI_100 dataset for the band selection, to verify the bands' portability. Changing the environmental conditions (from indoor to outdoor), the distance of acquisition (from 1.5 m to 5 m), the O_HSI_100 dataset (case (3), section 5.3) shows slight differences for what concern the PCA and the band selection. Indeed, Table 16 and Figure 53 show that, in this case the representative PCs component are two instead of three.

Table 16. Principal component analysis on a O_HSI_100 sample image: eigenvalues and cumulative variance in percentage.

Component	Percent %	Cumulative%
1	95.9838	95.98
2	3.2706	99.25
3	0.3515	99.61

As reported in Table 16, the first two PCs amount to the cumulative total variance equal to 99.25% (the value is even higher than the range between 80% and 90%), and for the Kaiser's rule, the variance exceeds the 1%. Moreover, in the screen plot, the elbow is formed at the second component (Figure 53). It is possible to achieve the same results for the other images of the dataset. As mentioned above, the spikes in the eigenvalues plot (Figure 54) allow to identify the significative bands for what concerns the band selection.

Table 17. O_HSI_100. Significative bands in a representative test extracted from the two PCs.

Principal Component	Test1
1	14-30
2	14-30

In this case, the significative bins are located only in the first part of the spectrum from 14–30 (from 558.26 to 621.78 nm) (Table 17). Even if representative bands are 16 instead of 27, the part of the spectrum was recurrent compared with the I_HSI_100. To appreciate the differences in the accuracy assessment of the algorithms, however, the reduction of this dataset was performed using the same bands extracted by the previous analysis.

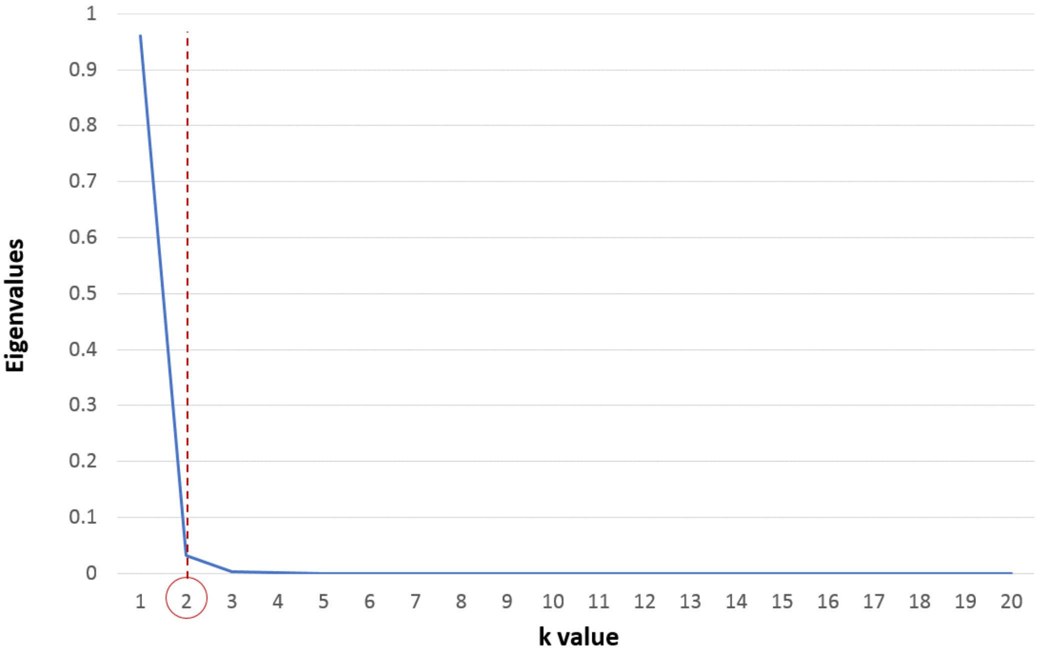


Figure 53. O_HSI_100: screen plot and zoom of the *elbow* for a sample image.

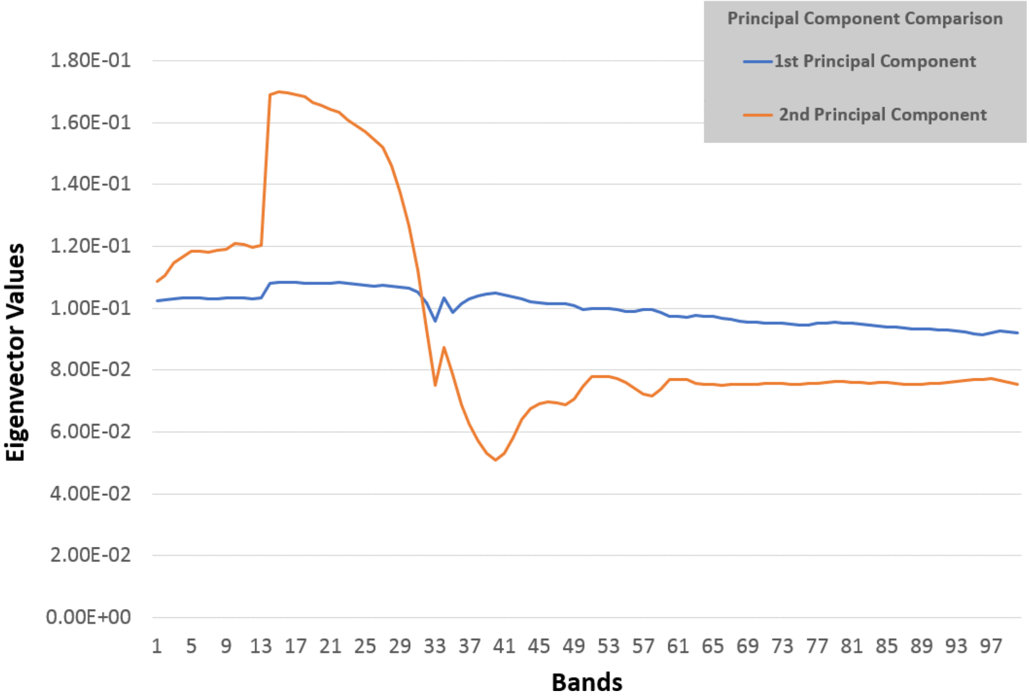


Figure 54. O_HSI_100: Eigenvalues plot for the first and second principal components.

5.5 Hyperparameter Tuning: Random Forest and Support Vector Machine

Hyperparameter tuning is an essential step in defining the classification model and has a critical impact on the accuracy of RF and SVM algorithms. The hyperparameter adjustment was made on the training set. However, the accuracy assessment was carried out for both the training and the testing set to verify the performances. The validation curve presents the model hyperparameters' values and shows the relation between hyperparameter and the accuracy trend. Overall accuracy, user's accuracy (training and the validation set) and the computational time were used as criteria for selecting the optimized hyperparameters for RF and SVM. In this case, the computational time has an important weight since the application requires a near real-time approach. In details, an accuracy-related focus on the rime ice, clear ice, and white aluminium were developed due to the importance that these three classes have in the real de-icing application.

The tuning was performed either for RF or SVM on the Test 1 image of both hyperspectral (I_HSI_100) and multispectral (I_MSI_3) datasets using the *Segmentation and Classification* toolbox of ArcGIS Pro 2.5.0 [299]. The hyperparameters estimated in this section were used for training and validation for the O_HSI_100 and O_HSI_10. The employed window workstation features are Windows 10 operating system, Intel® Core™ I7-6500U CPU at 2.50 Ghz, 16 GB of RAM, GPU AMD Radeon™ R7 M360 (Iceland) (six compute units at 980 MHz, 2048 MB).

A manual mode was chosen for hyperparameter optimization. Before the hyperparameters adjustment, data were split in 80% for training and 20% for testing. For what concerns the RF algorithm (section 5.5.1), the tree depth and the number of trees were tuned, fixing the maximum number of samples for each class. Instead, for the SVM (section 5.5.2), the maximum number of samples per class was changed.

5.5.1 Random Forest: hyperparameter tuning

For tuning the Random Forest model of hyperspectral data (I_HSI_100), the sample size was 2000 for each class. The tree depth values were set varying from 5 to 30, fixing the number of the trees equal to 50. As presented in Table 18, the difference between the overall accuracy is comparable in all training and validation configuration. Focusing on the validation outcomes, the best value of OA and glaze ice accuracy corresponds to the tree depth of 30. The low number of trees makes the case with a depth equal to 5 not significative. Therefore, the selected tree depth is equal to 30. For what concern the training computational time, it is possible to notice that the increase of the depth number does not influence excessively the speed of the process.

Table 18. RF: tree depth tuning. Training, validation accuracy and processing time for the I_HSI_100 dataset. R_i refers to the user's accuracy of rime ice, C_i to clear ice, and W_a to white aluminum. In bold, the selected parameter.

D_Trees	Training Accuracy (%)				Validation Accuracy (%)				Processing Time
	R_i	C_i	W_a	Overall	R_i	C_i	W_a	Overall	
5	73.68	91.06	74.75	92.54	42.48	92.96	64.89	85.54	34”
10	80.19	98.86	92.94	97.10	44.50	94.93	69.51	87.72	31"
15	78.80	98.85	93.45	96.90	43.13	95.01	68.96	87.06	26"
20	84.65	93.96	93.85	97.25	41.27	94.11	68.79	86.68	26"
25	79.34	97.14	93.49	96.85	42.23	92.52	68.93	86.52	27"
30	**81.60**	**99.42**	**95.38**	**97.25**	**43.69**	**95.78**	**70.87**	**87.83**	**33"**

For selecting the number of trees, the results reported in Table 18 taken into consideration. Indeed, the tree depth was fixed to 30. Table 19 presents the accuracy achieved with the number of trees ranging from 5 to 50. As mentioned for the selection of tree depth, the first analysed case can be considered not representative. The cases 15, 30, and 50 presents a similar OA; the highest C_i accuracy is achieved in the case 50. Analysing the processing time, the higher the number of trees, the higher the computational time is. Nonetheless, the computational time becomes stable after 15 N_tree. For this reason, the 50 N_trees was the chosen configuration.

Table 19. RF: number of trees tuning. Training, validation accuracy and training processing time for the I_HSI_100 dataset. R_i refers to the user's accuracy of rime ice, C_i to clear ice, and W_a to white aluminum. In bold, the selected parameter.

N_Trees	Training Accuracy (%)				Validation Accuracy (%)				Processing Time
	R_i	C_i	W_a	Overall	R_i	C_i	W_a	Overall	
5	85.34	96.59	91.01	97.15	39.09	93.95	66.89	86.02	14"
10	83.33	99.42	95.48	97.55	41.19	93.93	67.74	86.66	26"
15	82.38	99.42	94.25	97.40	45.47	95.16	71.37	87.96	32"
20	80.75	98.31	95.27	97.25	43.26	94.21	71.21	87.26	31"
25	85.86	97.21	95.05	97.70	42.86	93.50	72.03	87.33	32"
30	82.69	97.70	93.22	97.20	44.05	94.35	71.15	87.87	29"
35	86.87	99.43	95.65	98.00	44.08	93.75	70.89	87.34	33"
40	83.08	96.49	90.32	96.95	42.12	94.30	68.80	86.60	30"
45	81.64	98.85	92.13	97.10	42.64	95.10	69.97	87.44	33"
50	**81.60**	**99.42**	**95.38**	**97.25**	**43.69**	**95.78**	**70.87**	**87.83**	**33"**

Figure 55 reports the comparison between the validation curves obtained varying depths (above) and numbers of trees (below) respectively for the training and validation. The graphs confirm the previous conclusions described for the selection of hyperparameters.

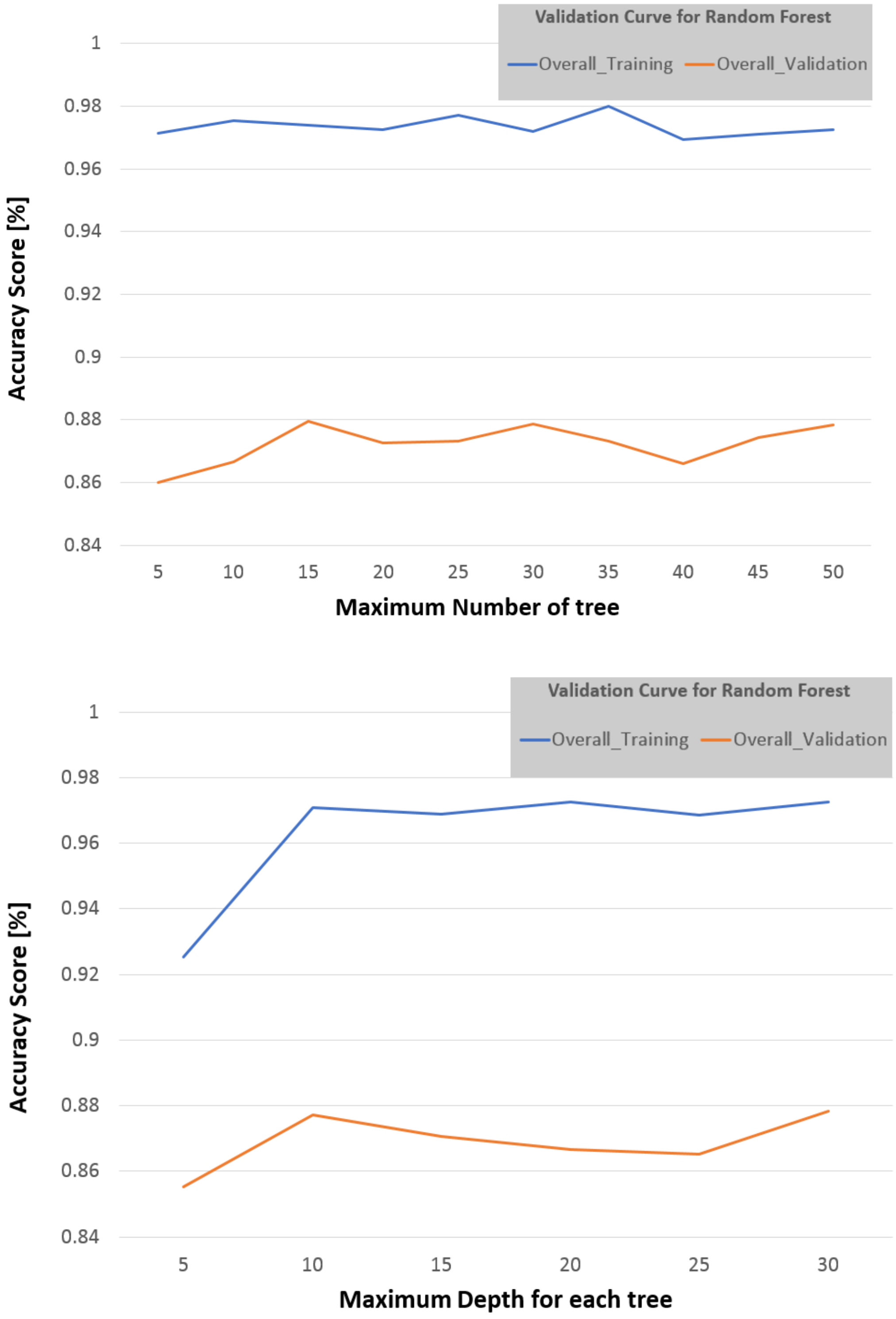


Figure 55. RF validation curve for I_HSI_100 dataset: maximum number of trees (above) and maximum depth (below) [298].

The model tuning for the multispectral data (I_MSI_3) was performed fixing the samples size equal to 10,000 for each class. As explained before (section 5.3), the number of samples is related to the image resolution. The image resolution also influenced the definition of the tree depth and the number of trees ranges. The starting configuration, the case 50_30, was chosen according to the tuning outcomes on the hyperspectral dataset. The tree depth ranged from 30 to 60, and the number of trees was set varying from 50 to 125. All combinations of the number of trees (*xx* in the test code) and the depth tree (*yy* in the test code) were evaluated, and the training and the validation accuracies ware reported in Table 20.

Table 20. RF: tree depth and the number of trees tuning. Training, validation accuracy and training processing time for the I_MSI dataset. The code xx_yy defines the test name, composed by xx number of trees and yy depth. R_i refers to a user's accuracy of rime ice, C_i to clear ice, and W_a to white aluminum. In bold, the selected parameter.

Test (xx_yy)	Training Accuracy (%)				Validation Accuracy (%)				Processing Time
	R_i	C_i	W_a	Overall	R_i	C_i	W_a	Overall	
50_30	70.51	66.92	56.34	81.68	55.37	68.76	60.25	77.13	2’59”
50_40	69.80	67.36	56.30	81.66	54.11	67.80	60.70	77.05	2’54”
50_50	69.56	67.59	55.83	81.53	53.10	68.42	60.89	77.29	2’57”
50_60	70.08	67.60	54.22	81.47	53.53	68.51	60.69	77.62	3’4”
75_30	68.83	67.24	55.58	81.43	53.95	67.95	60.43	77.13	4’50”
75_40	69.98	67.60	56.22	81.79	53.45	68.78	59.35	76.99	5’8”
75_50	69.57	66.33	55.02	81.57	53.02	67.96	59.43	77.06	5’4”
75_60	68.83	66.62	53.47	81.24	54.90	68.49	61.27	77.37	5’13”
100_30	**67.25**	**67.96**	**54.35**	**81.49**	**56.25**	**69.66**	**61.66**	**77.82**	**6’55”**
100_40	71.20	67.55	56.88	81.88	54.23	69.10	60.45	77.62	7’23”
100_50	70.63	67.46	57.58	81.31	54.51	68.71	61.10	77.38	7’25”
100_60	69.86	67.53	55.84	81.33	54.04	69.29	60.87	77.56	7’24”
125_30	68.86	66.37	54.75	81.21	53.22	69.24	59.25	77.45	9’23”
125_40	**69.04**	**67.59**	**55.73**	**81.72**	**52.59**	**69.85**	**60.45**	**77.62**	**8’58”**
125_50	69.58	67.84	56.74	81.73	53.66	69.14	59.27	77.33	9’1”
125_60	69.12	66.65	55.63	81.42	53.72	69.74	60.79	77.79	9’25”

All configurations present an OA value stables in training (81%) and validation (77%). This means that the hyperparameters variation does not conditionate the OA. As a consequence, it is possible to examine only the validation accuracy. Considering the clear ice UA, the best fit selection is also tricky, because the UA does not change too much in all configuration.

In this case, the computational time represents a decisive parameter to find the best set of hyperparameters. It is worth to underline that the processing time trend is increasing according to the number of trees rising. Considering these observations, the case 100_30 seems to be a trade-off between processing time and accuracy compared to all other configurations.

5.5.2 Support Vector Machine: hyperparameter tuning

To optimize SVM model for the hyperspectral dataset, the maximum number of samples per class was fixed between 100 and 5000 samples (Table 21). The highest value of the OA, considering the validation, results in the case of 5000 samples. However, this configuration was excluded due to high computational time (3h4'25"). Therefore, the processing time increases accordingly with the rise of the sample size. The case of 100 was discarded because the sample size was not big enough to compensate for the random choice. The risk, in this case, is that sample of some classes are not enough. The 750, and 1000 configuration, thus, were taken into account. Since the OA is comparable, and in the case of 1000, the time is still high, validation C_i accuracy was considered. The C_i is respectively 93.96 % in the 750 sample case and 92.67 % in the 1000 sample case. Looking at these considerations and the validation curves (Figure 56), the sample size equal to 750 represents the best fit.

Table 21. SVM: sample size optimization. Training, validation accuracy and training processing time for I_HSI_dataset. R_i refers to user's accuracy of rime ice, C_i to clear ice, and W_a to white aluminum. In bold, the selected parameter.

N_Samples	Training Accuracy (%)				Validation Accuracy (%)				Processing Time
	R i	C i	W a	Overall	R i	C i	W a	Overall	
100	77.67	97.74	92.17	96.40	56.69	99.17	72.47	91.47	2'8"
250	86.87	97.74	96.76	98.00	46.84	96.46	74.52	89.72	12'26"
500	86.67	1.00	94.62	97.95	44.71	95.28	73.17	88.62	24'51"
750	**93.51**	**99.44**	**98.46**	**99.05**	**51.86**	**93.96**	**90.91**	**90.92**	**47'21"**
1000	91.89	98.90	97.38	98.75	52.86	92.67	89.00	91.07	1h18'2"
2000	91.53	1.00	97.88	99.00	49.68	93.87	89.44	89.82	1h33'36"
5000	97.19	1.00	98.49	99.60	52.98	92.81	91.19	91.37	3h4'25"

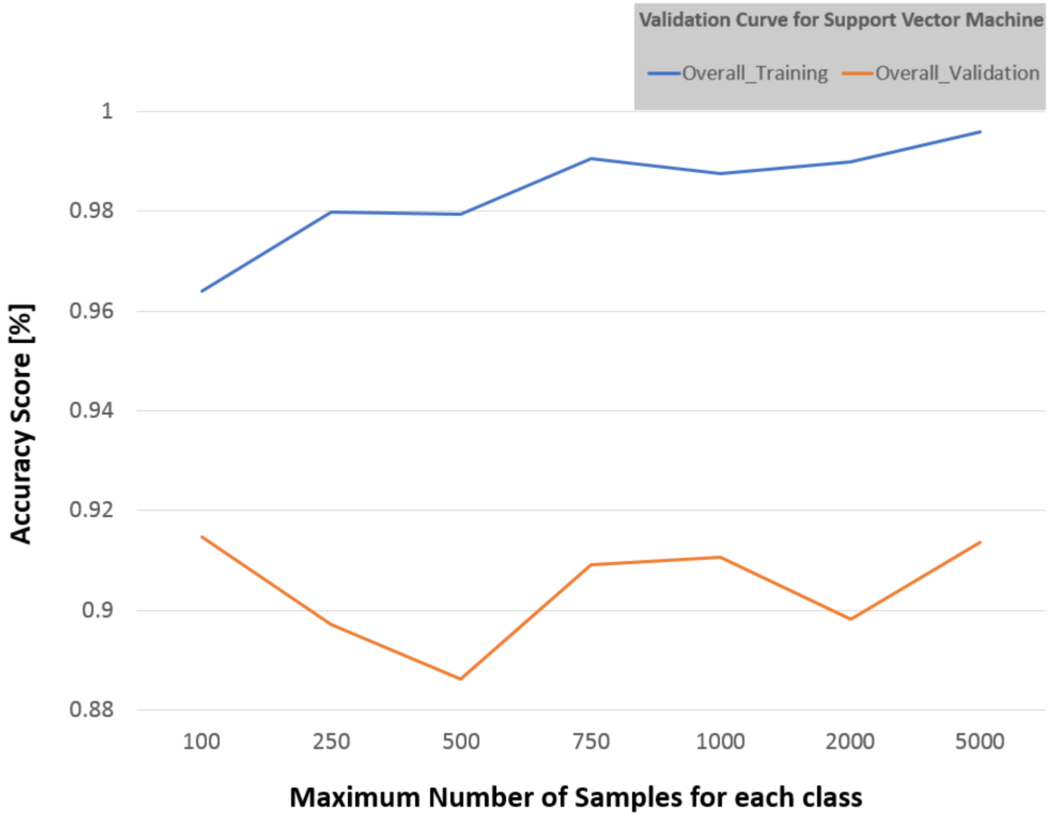


Figure 56. SVM validation curve for the I_HSI_100 dataset: sample size.

For the multispectral dataset (I_MSI_3), the number of samples per class was varied between 500 and 2000. Table 22 shows that cases with 1500 and 2000 samples have the best OA. In details, the latter (2000 samples) also has the highest value for the C_i accuracy (68.6%). In general, looking at the processing time, it increases according to sample size in all configurations. However, comparing the test with the best OA, the test with 1500 samples takes around 20 min less than the test with 2000 samples (1h5'43"). This makes the 1500 samples the best fit.

Table 22. SVM: sample size tuning. Training, validation accuracy and training processing time for I_MSI dataset. R_i refers to the user's accuracy of rime ice, C_i to clear ice, and W_a to white aluminium. In bold, the selected parameter.

N_Samples	Training Accuracy (%)				Validation Accuracy (%)				Processing Time
	R_i	C_i	W_a	Overall	R_i	C_i	W_a	Overall	
500	67.82	62.16	54.31	80.92	51.92	64.77	60.34	75.95	24'47"
1000	67.12	64.28	57.50	80.46	51.82	66.78	62.06	76.61	50'47"
1500	**66.77**	**64.84**	**54.53**	**80.21**	**52.54**	**67.69**	**61.03**	**77.07**	**1h5'43"**
2000	68.62	66.60	54.03	80.28	53.97	68.60	61.35	77.05	1h23'20"

5.6 Ice Detection

After the hyperparameter tuning the ice detection was performed using hyperspectral and multispectral data. For analyzing the dimensionality reduction effect and pointing out the advantages of using hyperspectral data instead of the multispectral data, different datasets were examined. To identify of dataset, an encoding *x_yyy_zzz* was defined, where: *x* identifies the environmental conditions, *I* for indoor or *O* for outdoor, *yyy* stands for the type of data, *HIS*, *MSI* or *PC*, and *zzz* is the number of the bands. The datasets can be summarized as follows considering both the indoor and outdoor cases (section 5.3) and the different reduced hypercubes (section 5.4):

- Indoor:
  - I_HSI_100: the original hyperspectral images acquired with the Senop Rikola;
  - I_HSI_27: the 27-band hypercubes resulted by the PCA band selection;
  - I_HSI_10: the 10-band hypercubes resulted by the PCA band selection;
  - I_PC_3: the 3-PC hypercubes resulted by the PCA feature extraction;
  - I_MSI_3: the original data collected with the MAPIR (RGN images);
  - I_MSI_4: the RGBN multispectral image with the blue band;
- Outdoor:
  - O_HSI_100: the original hyperspectral images acquired with the Senop Rikola;
  - O_HSI_10: the 10-band hypercubes acquired with the Senop Rikola. As mentioned in section 5.3, in this case, the band sequence was defined according to section 5.4.

For the classification, Test_1 for the training and the *Classify Raster* tool of ArcGIS Pro 2.5.0 [300] was used.

As mentioned in section 5.3, even if some materials are included in the background, they were annotated as different classes to control the performance on different materials. Considering that the real case background is composed of pavement instead of floor tiles, some classes are not representative. Thus, the overall accuracy was reported to give an idea about the algorithms' general performance. Nonetheless, the main parameter is the user's accuracy, because the detection of the ice and in particular, the clear ice is the object of the study. For each dataset, listed above, RF (section 5.6.1) and SVM (section 5.6.2) with the tuned hyperparameters derived from section 5.5 were tested. Table 23 summarizes the selected hyperparameters for each dataset.

Table 23. RF and SVM selected hyperparameters. HSI dataset includes all the dataset derived from hyperspectral data (original hyperspectral images, reduced hypercubes with band selection and feature extraction approaches), MSI dataset includes the RGN images and the RGBN images.

Dataset	RF hyperparameters			SVM hyperparameters
	Sample size	number of trees	tree depth	Sample size
HSI	2000	50	30	750
MSI	10,000	100	30	1500

In this section, the analysis takes into account the accuracy and the processing time for the classification only. In general, as mentioned in the section before both overall accuracy and user's accuracy were evaluated for classification assessment. As in the PCA case (section 5.4), four representative images (Test_1, Test_2, Test_6, and Test_10) were considered for the classification assessment both for an HSIs and an MSI dataset. Varied environmental conditions characterize Test_1, Test_2 and Test_6. Test 10 was included to evaluate the model's behaviour in the presence of the ice phase transition. Indeed, it is the case in which the ice starts to melt. It is worth to underline that ice changes its features according to its state and the ice detection in other physical state is out of this study. However, the inclusion of Test 10 helps to generalize the models. For what concern the RGBN dataset, the Test_1 integrated with the blue band is described to show the accuracy enhancement related to the blue channel presence.

5.6.1 Results of ice detection using Random forest

This section presents the results of the ice detection using the random forest. A comparative analysis among the HSI datasets (laboratory and outdoor condition) and MSI datasets was performed. In Table 24, it is possible to notice that the overall accuracy decreases based on the reduction of the bands. Moreover, for all Test_10, the OA and UA are lower than the other test as the ice started to melt. Since it is the only image in which the ice has a different physical state, it is predictable that the algorithm can reach the worst accuracy.

Table 24. RF: accuracy and processing time on the HSI datasets (indoor and outdoor) and MSI indoor. R_i refers to rime ice, C_i to clear ice, and W_a to white aluminum.

Dataset	Test	Accuracy (%)				Processing Time
		R_i	C_i	W_a	Overall	
I_HSI_100	Test_1	41.73	96.33	69.09	87.38	13'22"
	Test_2	46.32	96.36	70.95	88.84	14'38"
	Test_6	53.24	81.64	69.12	87.00	12'37"
	Test_10	43.34	52.79	59.54	81.58	16'9"
I_HSI_27	Test_1	41.70	92.78	69.14	82.54	25"
	Test_2	52.22	93.28	74.64	86.06	28"
	Test_6	40.43	71.00	67.02	80.36	31"
	Test_10	34.18	54.30	56.05	76.38	30"
I_HSI_10	Test_1	43.52	95.44	69.92	80.90	24"
	Test_2	50.18	94.81	74.72	83.24	27"
	Test_6	38.99	80.38	59.73	77.02	26"
	Test_10	30.70	57.78	56.69	73.05	29"
I_PC_3	Test_1	37.82	94.82	65.92	75.86	34"
	Test_2	41.14	75.62	78.63	73.59	39"
	Test_6	33.59	68.31	57.97	69.41	39"
	Test_10	46.13	54.95	72.16	70.95	41"
I_MSI_3	Test_1	54.96	69.79	60.72	77.63	6'40"
	Test_2	36.18	42.58	35.29	47.92	6'36"
	Test_6	25.20	33.79	8.24	30.92	6'30"
	Test_10	34.71	33.38	44.81	41.34	7'5"
I_MSI_4	Test_1	57.99	54.23	83.72	84.50	4'49"
O_HSI_100	Test_1	-	93.06	86.34	96.01	15'11"
O_HSI_10	Test_1	-	86.28	64.61	84.27	56"

Table 25. I_HSI_100: Example of the confusion matrix (Test_1). The similarity among the white aluminium and rime ice: false-positive samples (in red).

Class	R_i	C_i	F_t	W	83 WRP	27 GRP	21 GRP	W_A	A	2 B RP	Total	UA [%]
R_i	108	3	0	0	0	0	0	148	0	0	259	41.73
C_i	0	231	0	0	6	0	0	2	0	0	240	96.33
F_t	0	0	527	0	0	0	0	0	0	0	527	100.0
W	0	2	0	6	0	0	0	0	0	0	10	62.50
83 WRP	0	4	0	0	140	0	0	0	0	0	145	96.96
27 GRP	0	0	2	0	0	144	0	0	2	0	148	97.57
21 GRP	0	0	0	0	0	0	119	0	0	0	120	99.33
W_a	57	21	0	0	0	0	0	175	0	0	254	69.09
A	0	0	0	0	0	1	0	0	27	0	28	94.37
2 BRP	0	0	0	0	0	0	0	0	0	270	270	100.0
Total	165	261	530	6	147	146	120	326	29	270	2000	0.00
PA [%]	65.38	88.36	99.47	93.75	95.64	99.18	99.33	53.81	91.78	99.70	0.00	87.38

Looking at the difference between the I_HSI_100 case and the I_PC_3, the OA decreasing on average is 14%. The C_i user's accuracy achieves high values (96% on average) compared to the UA of the other classes. Focusing on the rime ice, its accuracy is around 42% since the spectral response of white aluminium and rime ice is similar. The example of the confusion matrix presented in Table 25 shows the similarity between the white aluminium and rime. Indeed, the rime ice and the white aluminium reported the highest number of false-positive samples (in red) compared with the other classes. Figure 57 and Figure 58 shows the results of the segmentation process on the indoor datasets. Also, on the visual results, it is possible to notice in all cases, the number of false-positive samples that belong to the rime ice class.

For what concern, instead the computation time, the trend is not clear considering the decreasing of the band number. What is expected is that the reduction in the number of bands produces a decreasing computational time. Even if the computation time seems similar in I_HSI_27 and I_HSI_10, the data shows a reverse trend using the I_PC_3 images. Therefore, the processing time is higher compared with the case in which I_HSI_27 and I_HSI_10 datasets were analysed. A possible motivation for this behaviour can be related to the lack of representative features that help material discrimination. The algorithm without the elements requires more time for returning the classification.

The accuracy assessment of the model using I_MSI_3 images presents that the overall accuracy has an average value of 49.5%, and the computational time is 6'43" in mean (Table 24). The presence of the blue band, I_MSI_4 produces an increase of both overall accuracy and user’s accuracy. However, the accuracy obtained is not comparable with the HSI results. For example, comparing the C_i user’s accuracy of the I_MSI_4 with the average C_i UA in the case of I_HSI_PC3, that are 54.23 % and 73. 43% respectively, it is possible to notice that also the I_MSI_4 cannot be considered accurate enough.

For what concerns the hyperspectral datasets in outdoor conditions, in both the O_HSI_100 and the O_HSI_10 case, the algorithm presents a processing time longer than the previous analysis (Table 24). As representative measures, Table 24 reports the white aluminium and clear ice accuracy. The overall accuracy for the O_HSI_100 is higher than the O_HSI_10, and in general user's accuracy is too. As it is possible to notice in Figure 60, in general, the segmentation of the O_HSI_100 is less noisy than O_HSI_10. However, clear ice is well detected in both cases. It means that, in general, outdoor conditions help algorithms perform better classifications as we expected.

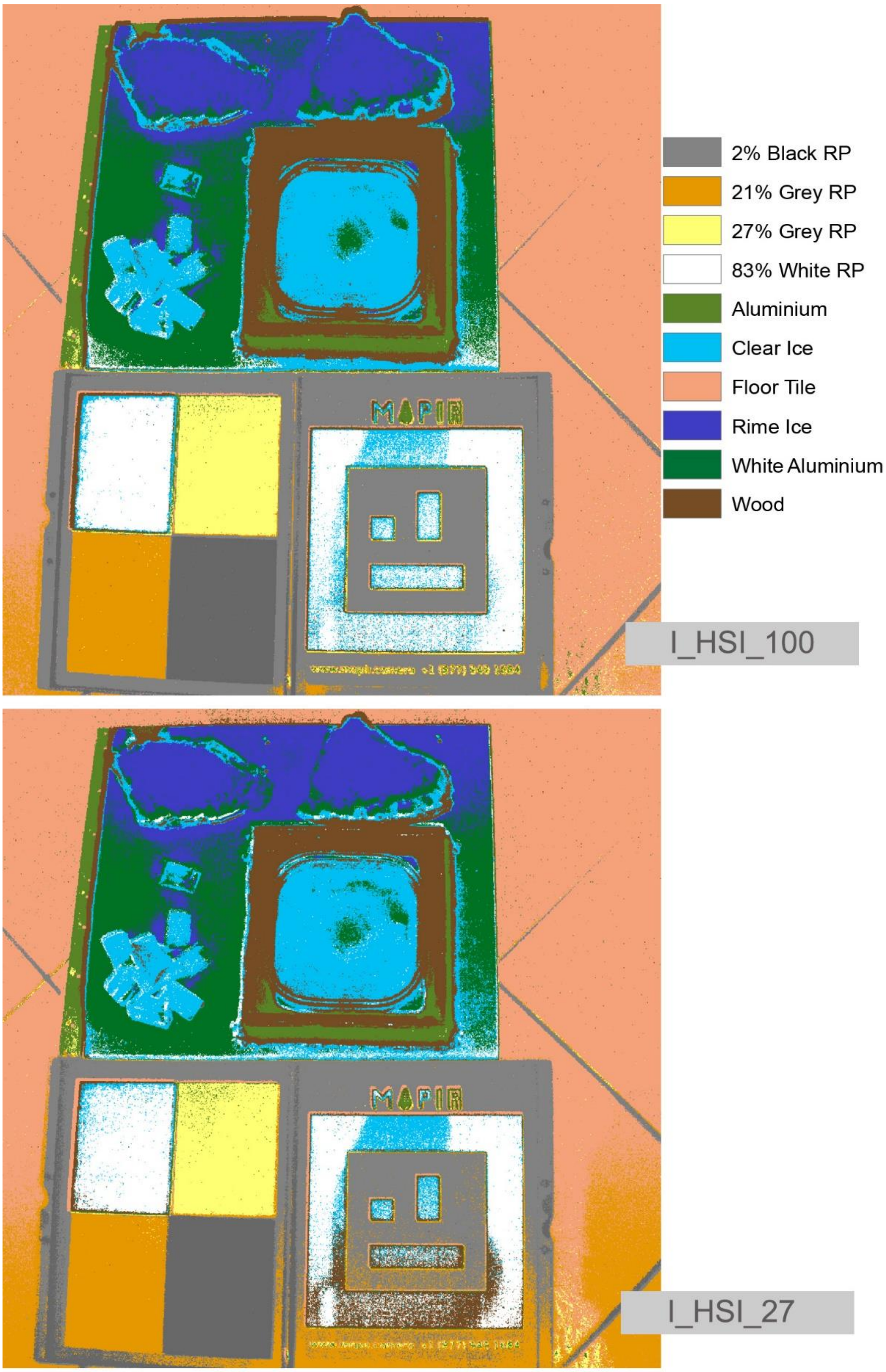


Figure 57. I_HSI_100 and I_HSI_27. RF classification outcomes on Test_1.

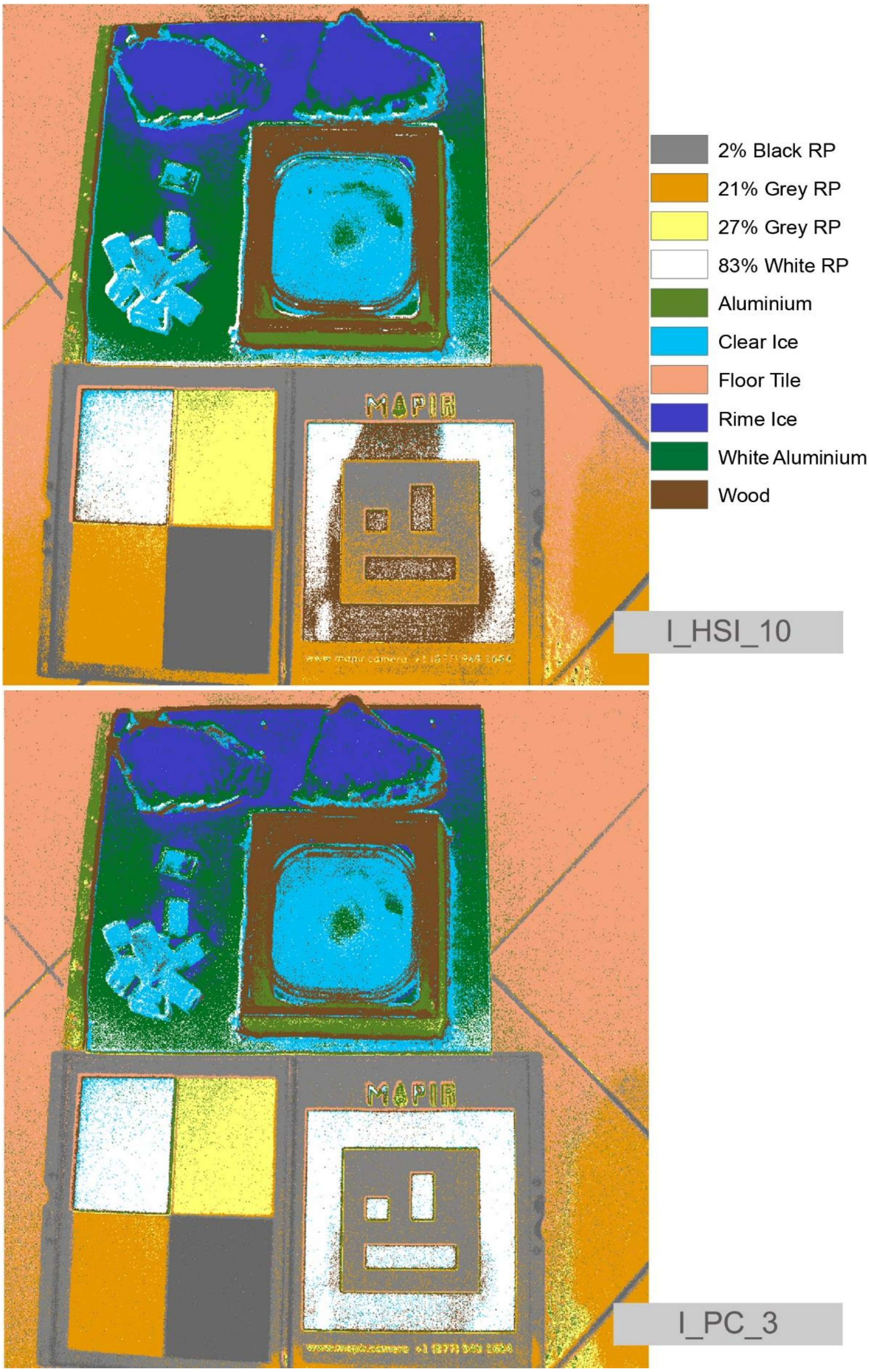


Figure 58. I_HSI_10 and I_PC_3. RF classification outcomes on Test_1.

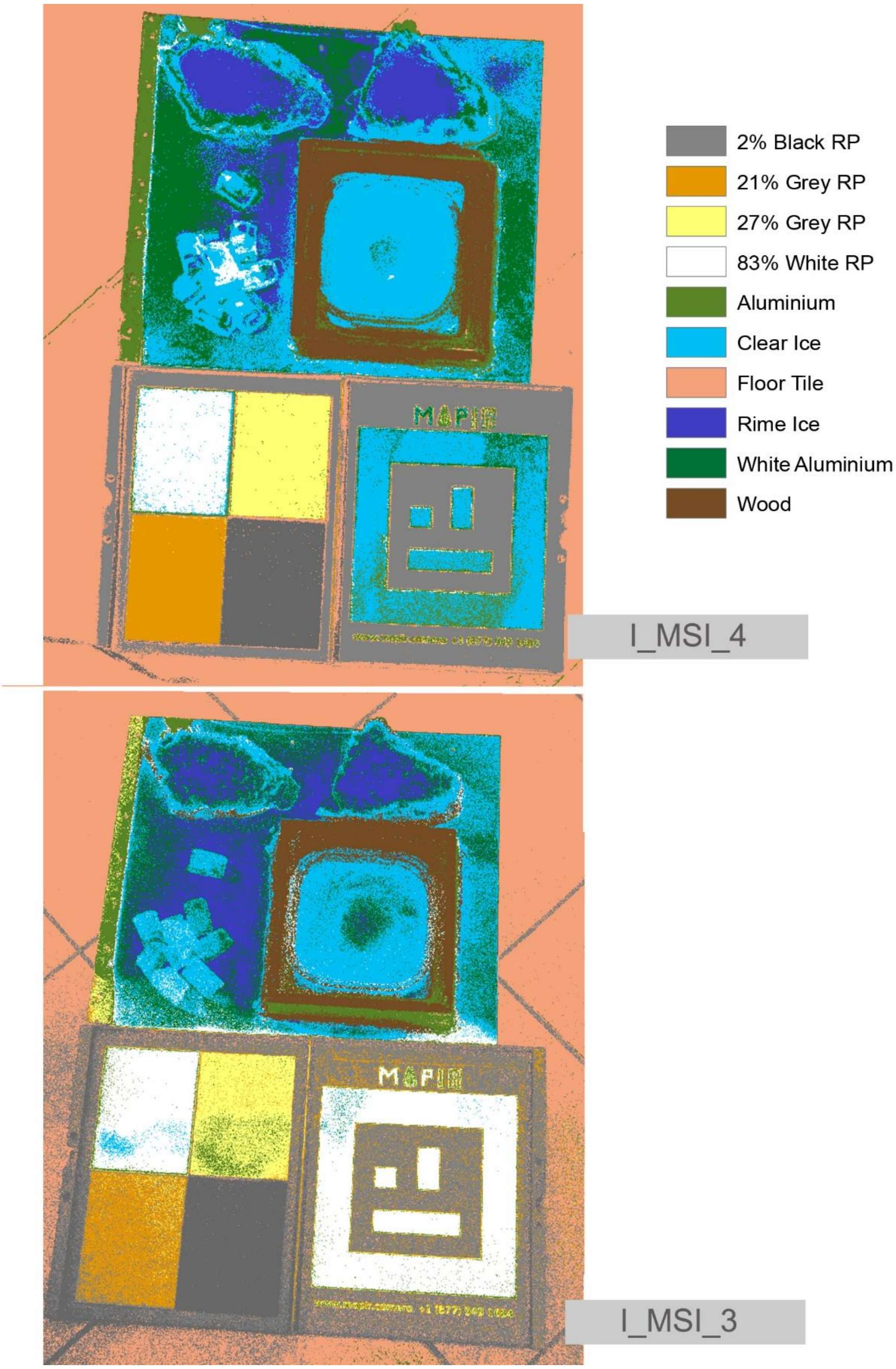


Figure 59. I_MSI_4 and I_MSI_3. RF classification outcomes on Test_1.

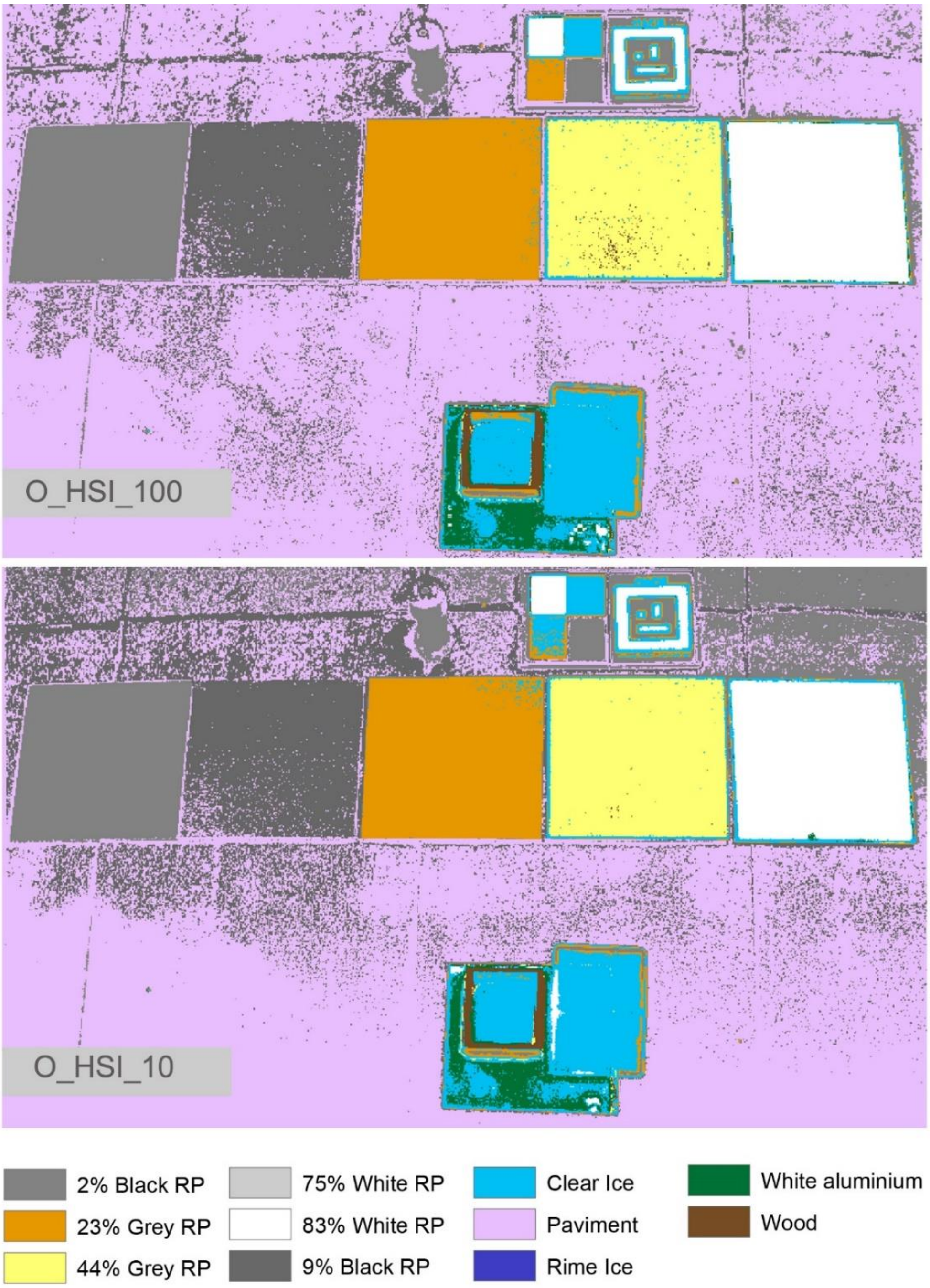


Figure 60. O_HSI_100 and O_HSI_10. RF classification outcomes on Test_1.

5.6.2 Results of ice detection using Support Vector Machine

The same datasets, in the same order, were classified with SVM classifier. Table 26 shows the accuracy and the processing time related to the HSI dataset analysed with the SVM. It is possible to notice that the overall accuracy of the I_HSI_100 achieves a maximum value of 92%, with a computational time around 17 min (Test_2). Moreover, in this case, clear ice user's accuracy (97%) is higher than the other significant classes. Evaluating the reduced hypercube (I_HSI_27, and I_HSI_10), the OA presents a maximum value of 87.3% and 80.8%, respectively for the I_HSI_27 and I_HSI_10. The processing time varies from 1'50" for the 27 bands to 1'40" for the 10 bands. Looking at the clear ice accuracy, it seems that the dimensionality reduction does not influence the results. Indeed, for both reduced datasets, the SVM has a higher user's accuracy of clear ice class, and it is slower than RF. Considering the I_PC_3, the average OA does not exceed the value of 76% and the processing time is 1'17" on average.

In general, the OA decreases, reducing the dimensionality like in the RF case (section 5.6.1). The rime ice accuracy maintains the same trend underlined in the RF assessment because its accuracy is on average lower than 60%. The Test_10 user's accuracy values are lowest compared with the other test in the same dataset as for the RF analysis. This evidence also recurs for all the indoor HSI datasets.

Table 26. SVM accuracy and processing time on the HSI datasets (indoor and outdoor) and MSI datasets indoor. R_i refers to rime ice, C_i to clear ice, and W_a to white aluminium.

Dataset	Test	Accuracy (%)				Processing Time
		R_i	C_i	W_a	Overall	
I_HSI_100	Test_1	50.49	95.2	88.03	90.39	15'2"
	Test_2	56.68	97.08	82.53	91.91	16'48"
	Test_6	56.58	80.3	72.59	88.39	15'4"
	Test_10	63.12	55.41	64.42	84.54	21'29"
I_HSI_27	Test_1	45.22	94.42	70.18	82.84	1'33"
	Test_2	62.48	96.79	72.62	87.32	1'52"
	Test_6	53.36	84.02	68.29	82.18	1'58"
	Test_10	45.74	61.01	59.35	78.02	1'57"
I_HSI_10	Test_1	44.12	94.97	70.04	79.37	1'35"
	Test_2	48.85	96.14	72.91	80.84	1'43"
	Test_6	49.37	84.38	67.85	78.89	1'42"
	Test_10	33.64	61.46	55.38	72.78	1'40"
I_PC_3	Test_1	42.50	95.39	68.25	78.16	59"
	Test_2	42.77	81.66	80.79	78.24	1'5"
	Test_6	45.16	77.89	64.81	76.05	1'16"
	Test_10	45.92	56.56	71.49	72.29	1'48"
I_MSI_3	Test_1	52.28	67.36	63.13	76.94	22'13"
	Test_2	34.84	41.27	32.86	46.85	23'9"
	Test_6	25.85	35.63	10.69	30.61	23'55"
	Test_10	40.31	37.14	42.46	42.38	25'21"
I_MSI_4	Test_1	62.37	57.38	88.15	86.70	8'18"
O_HSI_100	Test_1	-	97.82	93.17	94.11	18'26"
O_HSI_10	Test_1	-	89.11	54.69	79.28	2'38"

The results that concern O_HSI_100 shows that the algorithm presents a computational time longer than the previous analysis (Table 26). In general, the Overall accuracy and user's accuracy are higher than the I_HSIs.

Finally, the accuracy assessment of the I_MSI_3 dataset shows that OA is on average 49.2%, and the processing time is on average is 23'40" (Table 26). It is woth to notice that I_MSI_3 accuracy is not comparable with the ones obtained using the HSI datasets, even if to the I_PC_3, that has a similar feature space size. Moreover, the clear ice accuracy does not overcome the value of 67% in the best configuration. Considering the accuracy and the processing time in the I_MSI_4, instead, it is possible to underline that the OA increase compared with the I_MSI_3.

The OA is 86,70% for the I_MSI_4 instead of 76.94% in the I_MSI_3 best case. Moreover, what can be clearly seen is that the UA has the same trend as the OA. For what concerns the processing time, the SVM applied to I_MSI_4 is faster than the I_MSI_3.

Figure 61, Figure 62, Figure 63, and Figure 64 illustrate the SVM segmentation outcomes for each dataset, HSI indoor, MSI dataset and HSI outdoor, respectively. As it is possible to notice, in Figure 61, Figure 62 (I_HSIs), it is evident the decreasing of the accuracy due to the dimensionality reduction, and they confirm that the clear ice detection performs better in all the cases, compared with the rime ice identification. The latter, indeed, becomes more difficult the more the bands are reduced. A similar consideration can be reported in Figure 64 (O_HSIs). However, in this case, the noise is related to the pavement classification, that is not very relevant for the real application. Considering the MSI datasets (Figure 63), the enhancement due to the presence of the additional band is clear. Indeed, the classification noise is contained as well as the detection of the materials is improved. Furthermore, the contrast between rime ice and white aluminium also shows an enhancement.

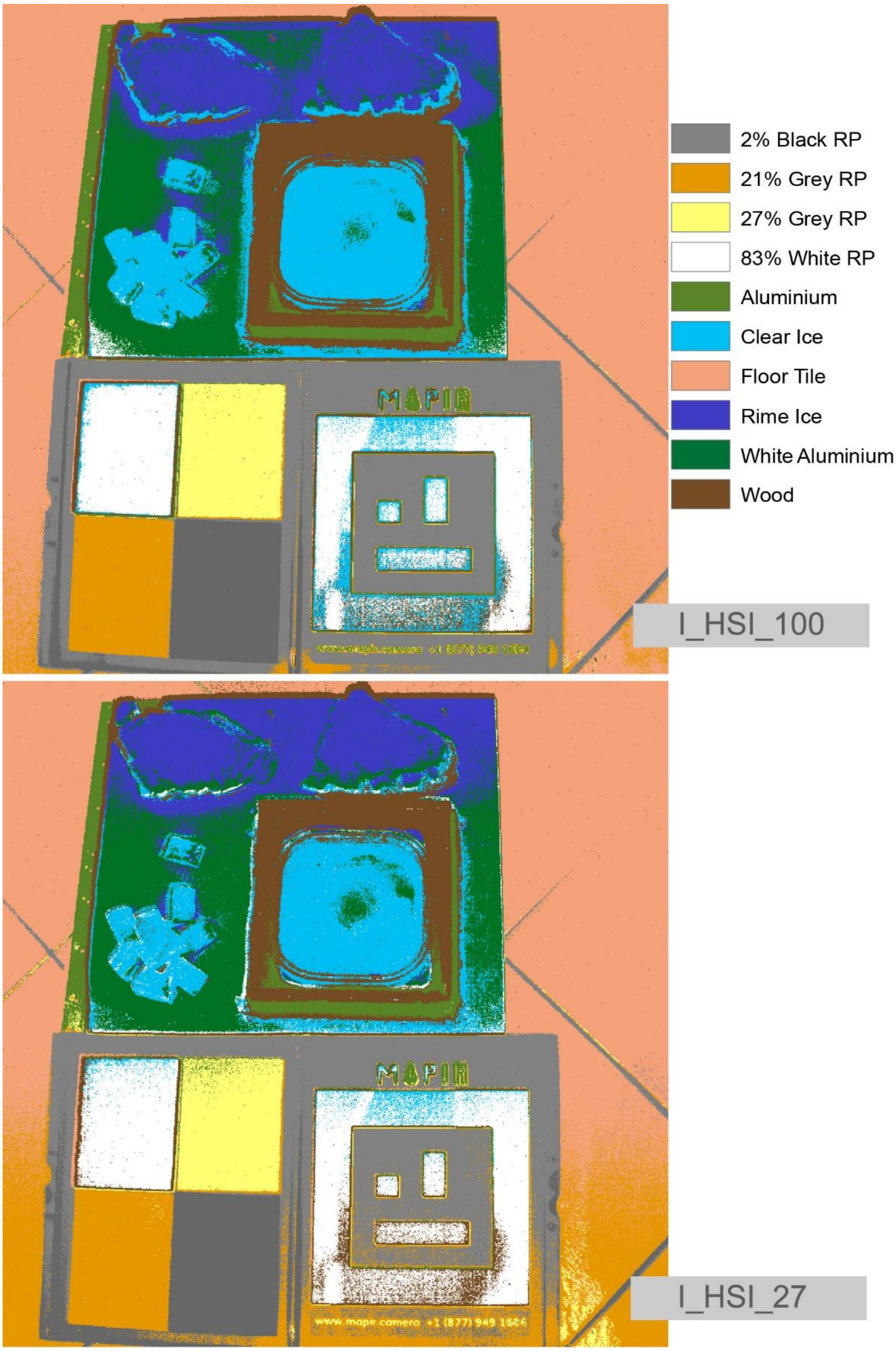


Figure 61. I_HSI_100 and I_HSI_27. SVM classification results on Test_1.

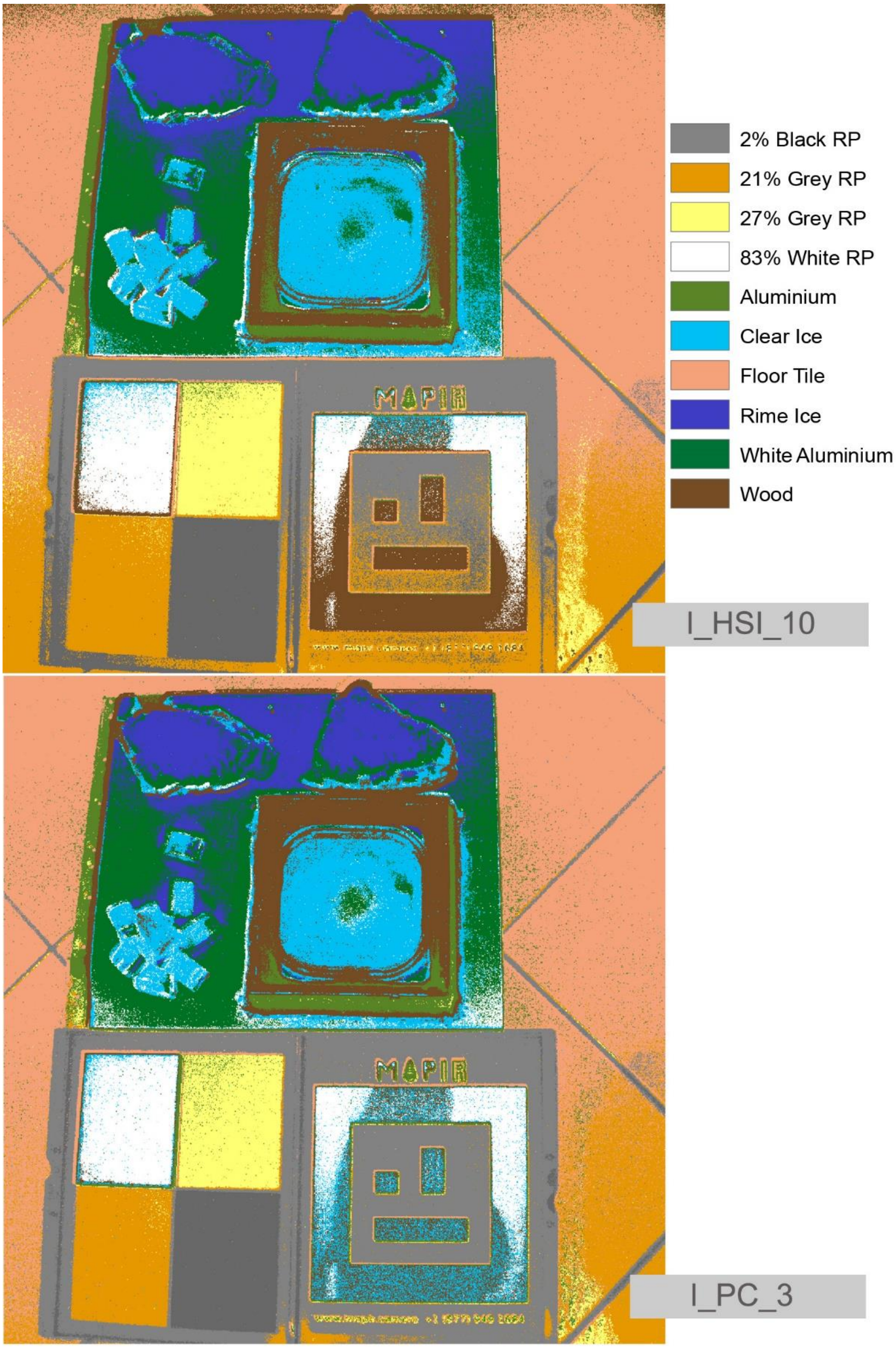


Figure 62. I_HSI_10 and I_PC_3. SVM classification results on Test_1.

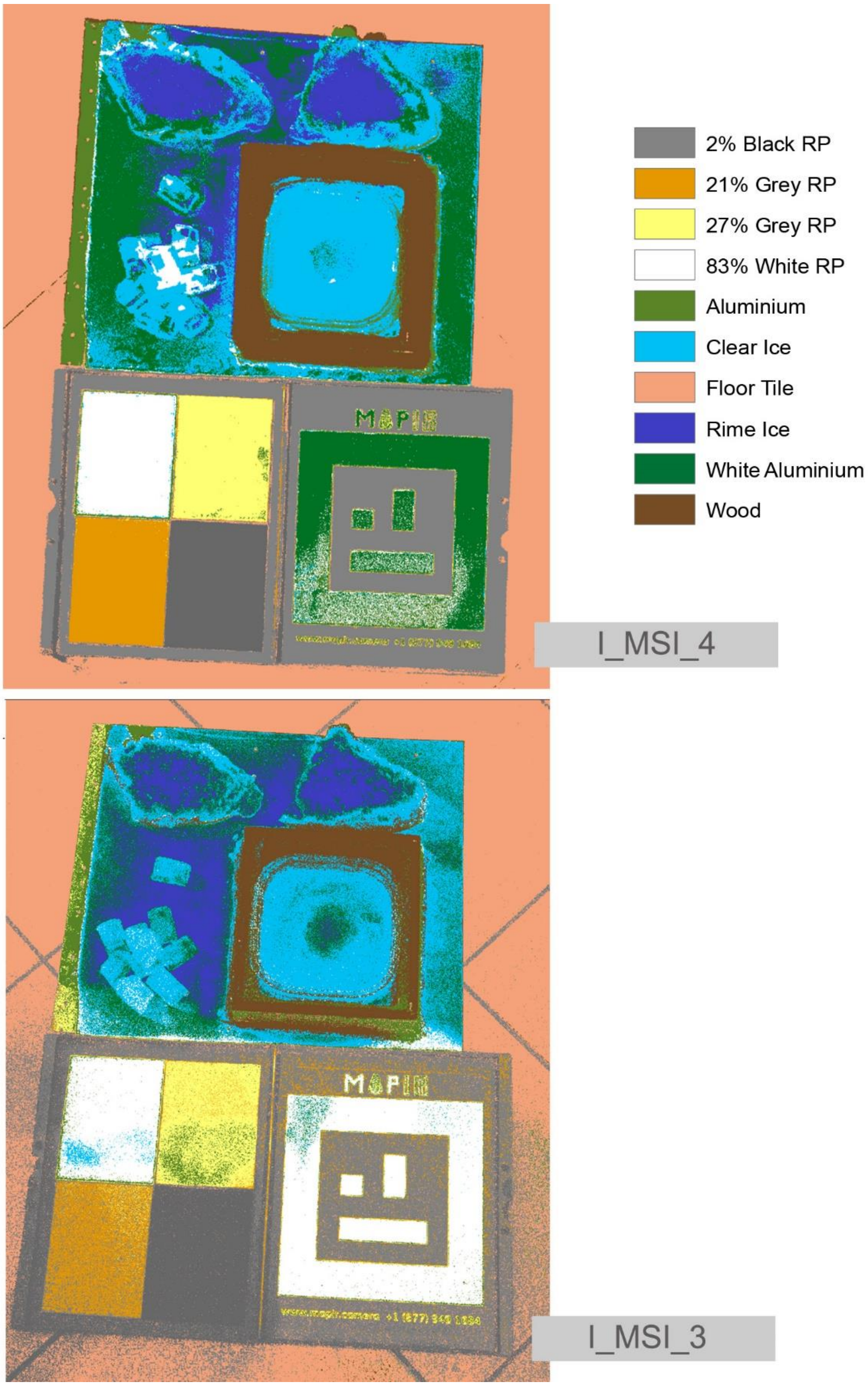


Figure 63. I_MSI_4 and I_MSI_3. SVM classification results on Test_1.

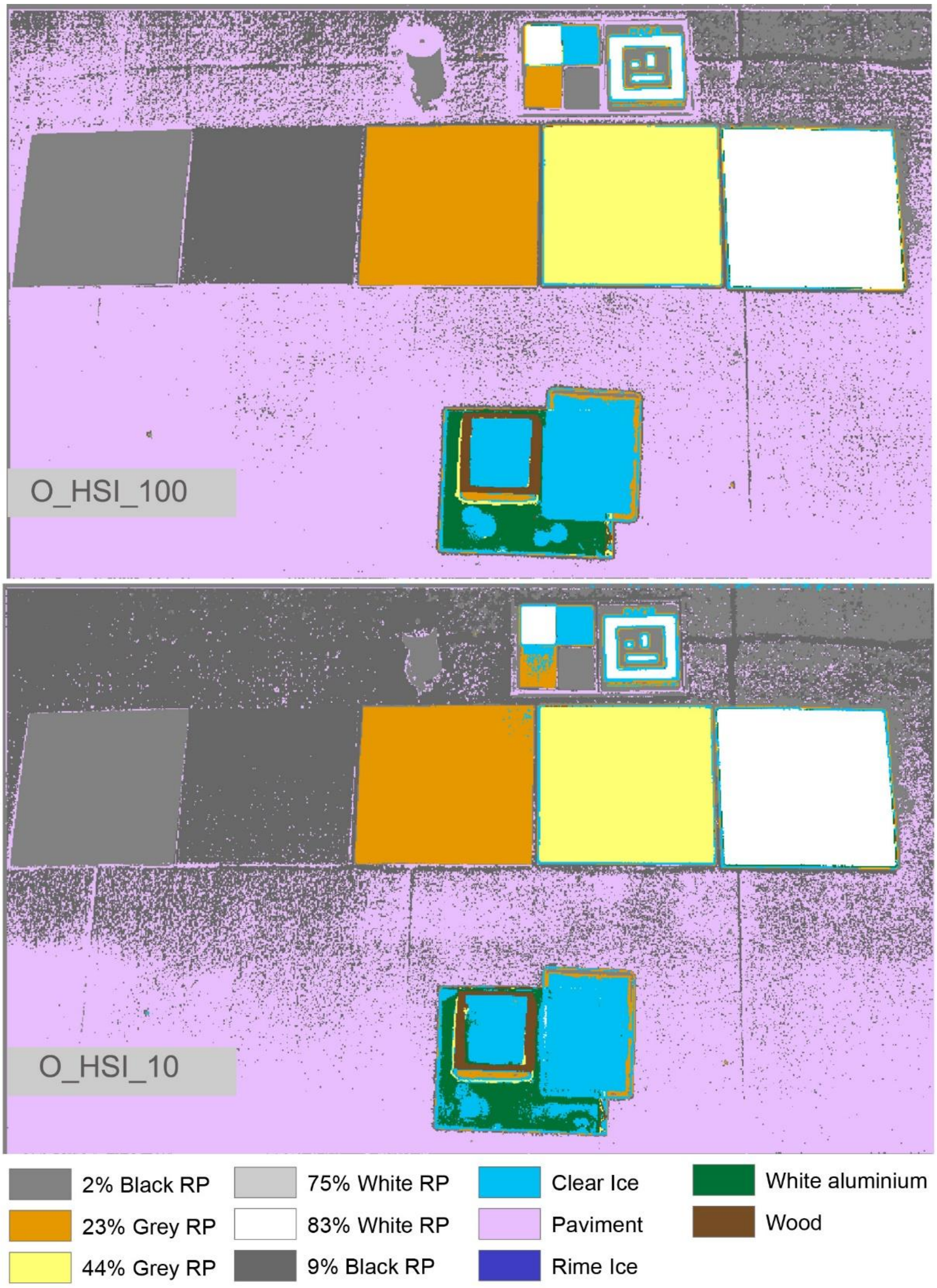


Figure 64. O_HSI_100 and O_HSI_10. SVM classification results on Test_1.

5.7 Discussion

According to the results of the previous sub-sections (5.6.1 and 5.6.2), the classification of the ice can be very challenging since this material can be invisible, i.e. clear ice, and can change its physical state, i.e., density and shape. Moreover, the classification of this type of ice, i.e. the clear ice, is not yet examined in the literature as far as the author is aware. Thus, this study represents the first attempt for its classification.

The outcomes show that the multispectral and hyperspectral data can be used for ice detection, even if there are significant differences in the accuracy and processing time. Thus, the segmentation results can be summarized according to these two criteria: accuracy and computational time.

For what concerns the accuracy, it is clear that hyperspectral data provides the best solution for the detection. Both the RF and SVM achieved the best accuracy in the segmentation using hyperspectral data, even reduced. Analyzing in-depth the single case in terms of accuracy, for HSI datasets, it is possible to underline that dimensionality reduction does not influence the overall accuracy strongly. SVM and RF accuracy are equivalent in all cases. For ice detection with HSI datasets, some general considerations can be presented. Despite minimal accuracy differences (from 0.1% to 2%), the user's and overall accuracy on average is higher for the SVM than the RF classifier.

Moreover, even if the identification of rime ice could seem more straightforward because it is white, it is not. The UA of the rime ice is in all the case lower than the other materials. This happens because its radiometric response is very similar to the aluminium. In the detection of the rime ice, the advantage is the possibility to improve the classification using spatial features, as mentioned in Aicardi et al. [301]. For what concerns the UA of clear ice, it is not visible a descending trend following the reduction of dimensionality. The discrepancies of clear ice accuracy between the best case (10 bands hypercube) and the worst case (PC images) are low. Indeed, they have values respectively of 7% for the SVM and 9% for the RF.

It is difficult comparing the accuracy achieved with the multispectral and hyperspectral data, regardless the selected algorithm because the data are different not only in for the spectral resolution, but also for the spatial resolution. However, what it is possible to recognize it is that both the dataset with three bands I_PC_3 (on average the OA is 74.31%) and I_MSI_3 (on average the OA is 49.32%), presents the lower overall and user’s accuracies. This means that three bands are not enough for the segmentation of such images. Indeed, improvements in the accuracy can be achieved with adding one single band, as in the case of I_MSI_4 (on average 85.6%)

Looking at the processing time analysis, it has to consider the previous analysis on the accuracy. The evaluation of the accuracy and the computational time reported in section 5.6 shows that the 10-bands case can be identified as a trade-off between accuracy and processing time, regardless of the employed algorithm. Therefore, OA is higher than 80% on average in both of the cases. However, the convenience to operate multispectral data is in the gain in term of the computational time.

Nonetheless, as a general consideration, regardless of the type of data, the tests highlight that the RF classifier is faster than the SVM. Considering the worst case, which means the HSI_100 classification, the average computational time is 14'22" for the RF and the 17'5" for the SVM. In contrast, the SVM is more accurate than the RF. Concerning the ice detection feasibility, it is clear from the results that both algorithms conservatively identify both ice types. Conservatively means that the algorithms recognize an area of aluminium as ice; the occurrence of the contrary situation is improbable. This is an advantage in the practical application because not spry de-icing fluids to the ice-contaminated area is more dangerous than to waste it. In the first case, indeed, the human safety can be compromised. According to this analysis, it is possible to identify the RF as a possible solution; however, object detection algorithms can be tested in further studies.

Besides the results and the implications of the dimensionality reduction in the segmentation, an outstanding outcome of the band selection, are the recurrence of the same original bands over time. The I_HSI and O_HSI analysis show that the significative bands for ice detection can always be selected to the same part of the spectrum, i.e. from 558.26 to 621.78 nm. This means that it is possible to assume the portability to these results on new datasets. The portability of the bands implies two further developments.

The first is related to the possibility to select an *ad hoc* multispectral sensors, designed for the specific application, exploiting the advantages of multispectral cameras. Based on the band selection, a multispectral sensor can be defined, improving the spatial resolution and reducing *a-priori* the spectral resolution. The second concerns the possibility of transfer learning and makes feasible using these massive data, such as hyperspectral, in near-real-time classification tasks. Transfer learning uses the knowledge acquired by the previous dataset to segmenting a new one. This allows to enhance the real-time procedure because the training time is significatively reduced.

Chapter 6

From semantic segmentation to object detection using RGB images

As for spectral images, dealing with data captured in different environmental conditions by different sensors, with different levels of details and multiple perspectives, can be challenging for the time-effective and efficient processing due to their heterogeneity. However, such types of data open up the possibility to improve the performance of the interpretation process in different applications. Moreover, besides the type of data, the information extraction process can be employed to perform different tasks, as explained in section 4.2. In some real-time applications, it could be useful detecting only the objects of interest using object detection techniques instead of discriminating all the objects in the scene through semantic segmentation.

The objective of this Chapter is to test the performances of these two different interpretation tasks, semantic segmentation and object detection, exploiting the use of RGB multi-sensory, multi-view, multi-resolution, and multi-temporal data collected by UAV in real conditions. Moreover, RGB original images have been used to take advantages of the full resolution, using two algorithms Random forest, for the semantic segmentation, and Faster R-CNN, for the object detection as they could fulfil the needs of big data management in quasi-real-time.

In this case, a precision agriculture-related application, such as plant disease detection, has been selected as representative. To understand the context, section 6.1 presents a brief description of the application and the related works.

6.1 Precision agriculture

Agriculture is one of the main economic sectors related directly and powerfully to the environment, economies, and societies [302]. While it has a primary activity with the credit to satisfy the food request, it has also caused environmental degradation. The environmental degradation concerns, first, the water resource availability. The agriculture sector is responsible for 60% and 90% of water resource consumption [303]. Moreover, substantial impacts on the environment due to agriculture are related to pesticide and fertilizer use, consumption of the soil and greenhouse gas emissions. Over the last few decades, due to the world population's constant growth, an unprecedented food demand has been further complicated the scenario, prompted an increasing interest in its sustainability. Indeed, for an economically and environmentally sustainable production system, there is a high request for efficient tools and techniques to find a trade-off between efficient agricultural production and reduced harmful environmental consequences. This demand has led to the development of the site-specific agricultural management system, that it is known as Precision Agriculture or smart agriculture. Indeed, Precision Agriculture refers to a farming system, integrated, based on information and production. It is designed to increase long term, site-specific and whole-farm production efficiency, productivity and profitability while minimizing unintended impacts on wildlife and the environment promoting the sustainability [56], [112]. Therefore, PA tries to cope with the spatial-temporal the agricultural system variability, using technologically advanced monitoring and automation tools. According to the Rural Development Program 2014-2020 [304], the main technologies and tools for the improvement of precision agriculture concerns:

- machine guidance: the automatic driving can avoid overlapping following the same tracks automatically for every field operation, driver relief, reduce chemicals and fuels [305];
- Geographical Information System (GIS) and decision-support systems: structured data storage can facilitate the exchange of information among different users, i.e. farmers, contractors, suppliers, and government for food security and tracking. Moreover, multi-layer datasets derived from automated data fusion of different sensor information can lead to real-time decisions;
- remote sensing techniques combined with machine learning techniques: the access to a different type of data analyzed through automatic techniques for information extraction allows to collect, visualize, and evaluate crop and soil health conditions at various stages of production conveniently and cost-effectively. In particular, they can help to estimate yield potential, nutrient deficiencies and stresses [306], [307], and can serve as an early indicator to detect potential problems [308], [309]

Over the years, the PA techniques have been most widely used in arable land, i.e. maize, sugar beet, soya, sorghum and rice, with the main purposes of (1) confining all field vehicles to permanent traffic lanes to minimize the crop damage and the fuel waste, (2) optimizing the use of fertilizers. On the contrary, the implementation of PA methods in orchard and vegetable farming is more recent than in arable land. In this case, the use of PA techniques allows to (1) monitor the food quality, to track the field operations, and (3) optimize the consumption of fertilizers and pesticides.

Among the orchards, the vineyard represents, nowadays, the main application field for smart agriculture. The implementation of PA concepts in viticulture, also known as Precision Viticulture, is driven by its significant economic relevance in the European Union (EU). The support of the technological and data-driven process promoted by the precision viticulture is related to the growth stage estimation and the potential quality of wine grapes [81] and early pest detection [310].

Since this sector uses the highest amount of pesticide and insecticide, there is a high request of technologies and techniques to contrast the plant diseases. Therefore, plant diseases always interfered with the production and influenced grape yield for the quantity and the quality [311]–[313]. According to Micheloni, 2017, the Flavescence Dorée (FD) among the vineyard diseases has a massive impact in the EU. FD is included in the A2 EPPO list (EC directive no. 2009/297EC) as a quarantined organism, and it is caused by a bacteria transmitted in the field by the leafhopper Scaphoideus titanus Ball [314]. The visual inspection-based techniques is the most common method used identification of this plant disease. It is a time-consuming approach that can be carried out on in the field or the laboratory, and it requires teams of experts [315]. Thus, it is clear that, especially for wide fields, the development of time-effective and efficient methods is the main challenge in this application field [316].

In this scenario, UAV (Unmanned Aerial Vehicle) remote sensing techniques combined with machine learning algorithms provide a great solution for rapid, non-destructive, cost-effective detection, and localization of the disease. Therefore, this approach allows the monitoring of plants and the detection of its biophysical, biochemical, and optical property changes of tissues and leaves through automatic analysis.

The use of automatic techniques, in plant disease detection, is particularly complex due to field conditions, outdoor environment, and the uncertainty regarding plant structure. Even the use of photogrammetric products such as orthomosaic, always adopted as input data for monitoring purposes [45],[46], becomes difficult. Indeed, in PA application where the background is homogeneous, and the presence of the trees is dominant, the information extraction results can be affected by geometric artefact. This is due to the feature extraction algorithms used for scene reconstruction (Figure 65). The use of the original images, employing a computer vision approach, can be adopted in this case to overcome this limitation and improve the results.

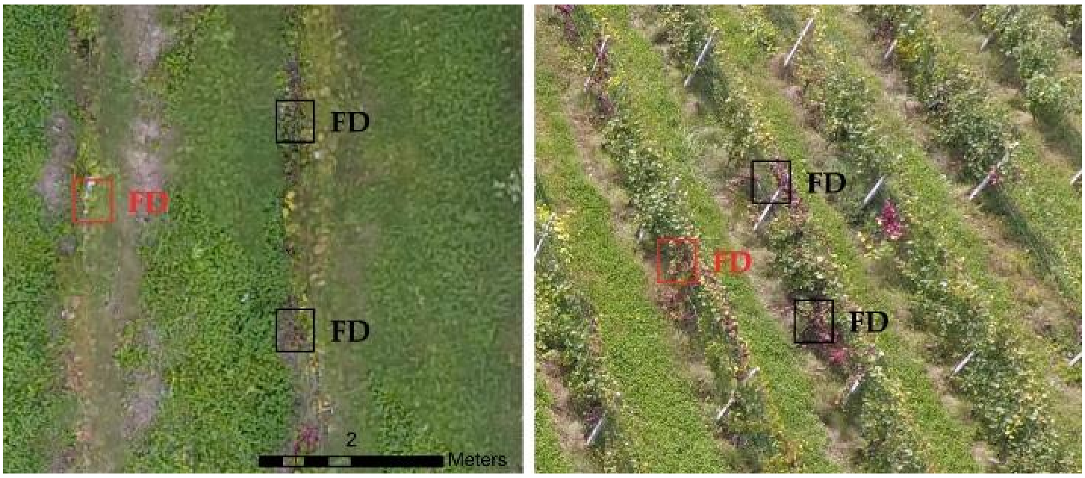


Figure 65. The influence of geometric artefacts, background homogeneity and the nadiral point of view. The comparison between an orthomosaic patch (on the left), and an original oblique image (on the right) in a vineyard [269].

Moreover, the development of automatic techniques for disease inspections is still in an early stage in this application field due to the lack of consistent datasets with a large number of labelled images acquired in real conditions (different lighting conditions, points of view, and with an inconsistent background) and due to the small size of the disease spots. Previous studies focused on the detection problem using semantic segmentation approach, which performs a pixel-wise image labelling. Machine learning algorithms, such as random forest, support vector machines, and deep learning techniques such as CNNs (e.g., AlexNET, ResNet, U-Net) have been employed for this purpose [319], [320]. The algorithms cited above were used to analyze the problem at different LoDs (from leaf to canopy scale). Previous works on leaf diseased used mainly datasets with images of individual leaves, and when UAV images were adopted, the detection was carried out on the orthomosaic [317], [321].

In this Chapter, the preparation of a dataset of UAV original imagery for the FD diseased plant at the canopy scale is presented to overcome the limitations. Moreover, the test of an object detection network and traditional pixel-wise classifier are presented.

6.2 From data collection to disease detection

For the experimental analysis, a vineyard in Baldichieri d'Asti (Piedmont, Italy) has been chosen as a case study. The area of interests, a property of the "Azienda Agricola Ciabot", is enclosed by forest and covers a hilly area of two hectares. The main feature of this vineyard is the presence of two grape varieties: Freisa and

Barbera (Figure 66). This entails an additional complexity in the disease identification because the several grapes imply different responses to the flavescence dorée. The flavescence dorée is, in both cases, characterized by leaves discolouration from greenish to reddish (Figure 67). However, the severity and the development of the disease change based on the types of grapes. The Freisa is more exposed to the disease than the Barbera; thus, the severity is higher.

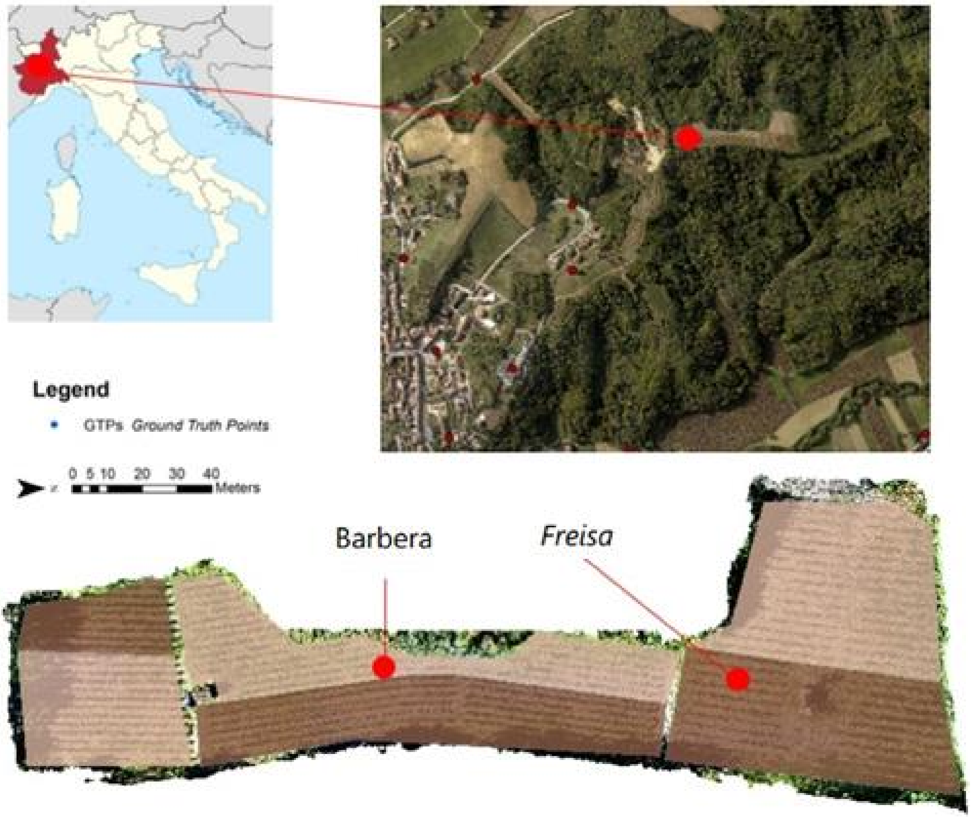


Figure 66. Azienda Agricola "Ciabot". An overview of the area (on the left). The vineyard thematic map (on the right). The Barbera grapes (in pink), the Freisa grapes (in brown) [269].

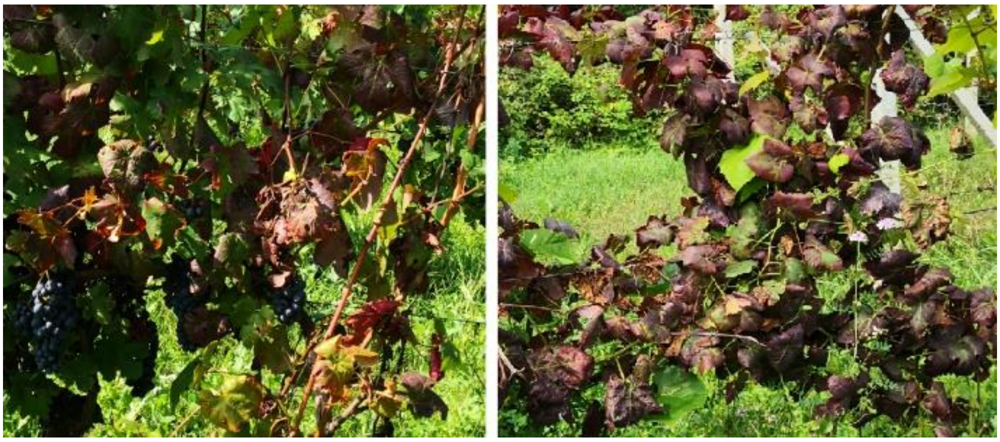


Figure 67. Example of an FD diseased plant (reddish leaves) [269].

Figure 68 reports the methodology workflow adopted for the analysis. It is possible to notice that there are recurrent steps compared to the analysis performed in Chapter 5. In details, they are the data collection, data annotation and preparation, detection and performance assessment of the two algorithms.

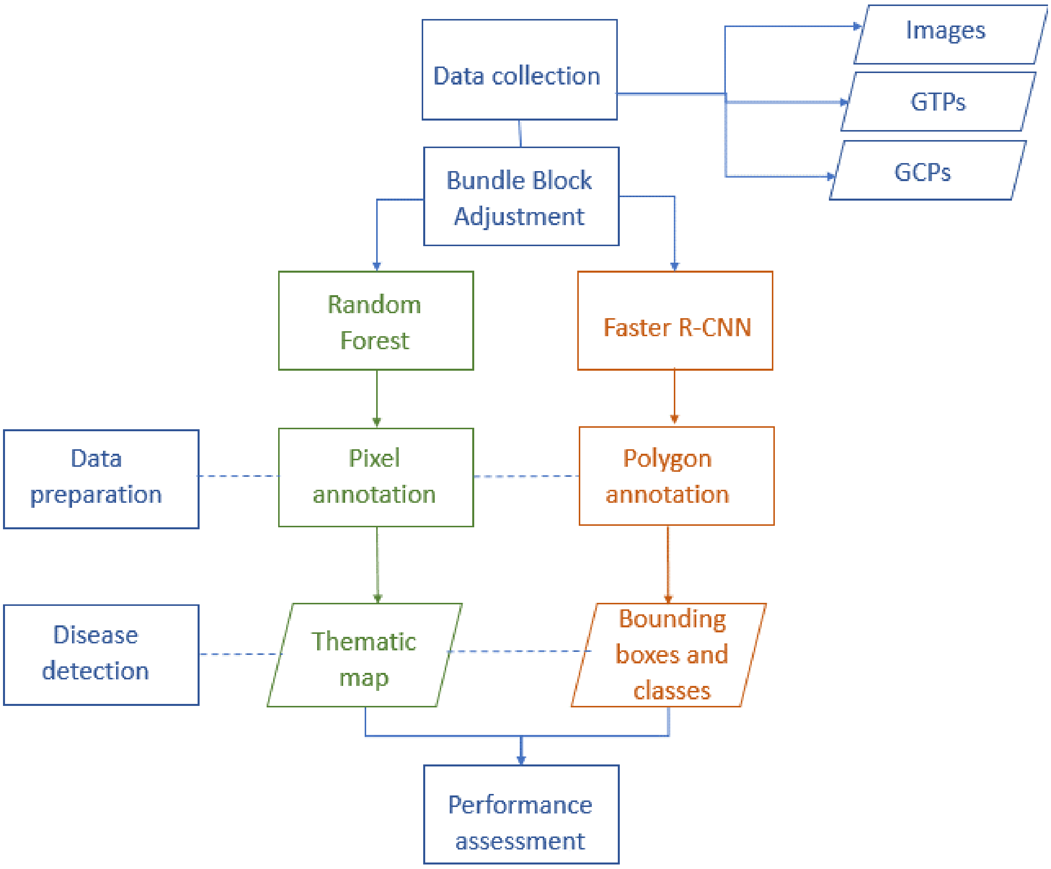


Figure 68. Methodology workflow. GTPs stands for Ground-truth points

The data collection was performed in September 2019. The period was selected according to the cycle of the disease. Indeed, in September, the symptoms of the disease are more evident. The survey was carried out using two platforms: DJI Phantom 4 Pro and the DJI Matrice 210 v2 equipped with RGB FC330 and DJI ZenMuse XT2 sensors respectively (section 3.1.1). As reported in Table 27 two different flight altitudes and patterns were considered to have multiple LoD (20 m and 25 m) and perspectives (nadir and oblique) (Figure 70).

Table 27. Data collection configuration: sensors, patterns, dataset sizes and GSD.

Imaging Sensors	Flight Height (m) and schema	GSD (cm/pix)	N of images
RGB: FC330	20 nadir	0.70	584
RGB: FC330	20 oblique (45°)	1.40	322
RGB: FC330	25 nadir	0.65	255
RGB ZenMuse XT2	25 nadir	0.65	255

Thus, a dataset of 522 images (resolution 4000×3000 pixels) was generated. At the same time, a field survey of the reference data (diseased samples) was carried out. Therefore, the coordinates of 20 well-distributed ground-truth points (GTPs) were measured through the Network Real-Time Kinematic (NRTK) – GNSS positioning [322]. The NRTK measurement was executed with a Trimble SP80 GNSS receiver, using real-time differential correction from the permanent GNSS station in Canelli (AT). Each reference point was registered and documented, as presented in Figure 69. Furthermore, the coordinates of 16 markers (GCPs) have been collected with the same techniques for georeferencing the images. The georeferencing procedure allows recovering the bounding boxes locations in a global reference system, after the object detection process on the original images.

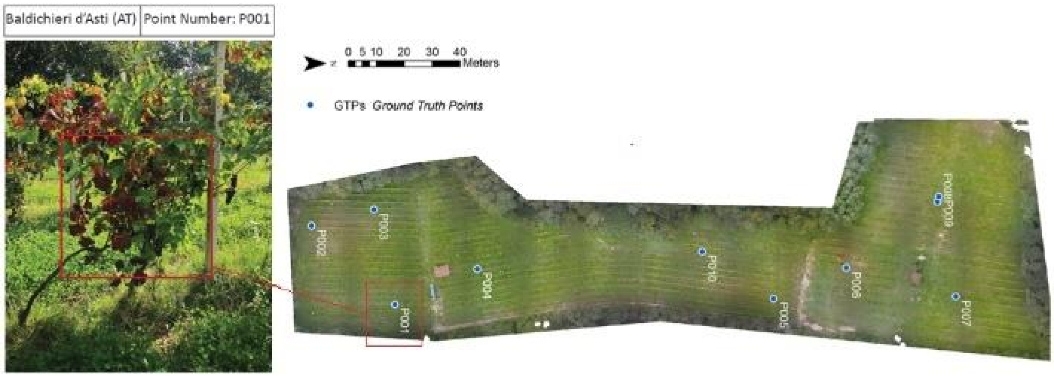


Figure 69. GTPs: an example of a form.

After the data collection, a subset of 200 images has been prepared for detection processing. Considering the acquisitions described in Table 27, fifty images for each dataset were selected to enclose as many cases as possible and scenarios with different resolutions, illumination conditions, points of views and homogeneous backgrounds. Figure 70 reports some examples. With this approach, both the data augmentation problem and the generalization capability of the models were addressed. Then the selected images were cut into smaller overlaid sub-images with a size of 1024x1024 pixels. This step is needed for preventing the resize preprocess accomplished by Faster R-CNN (section 4.2.2).

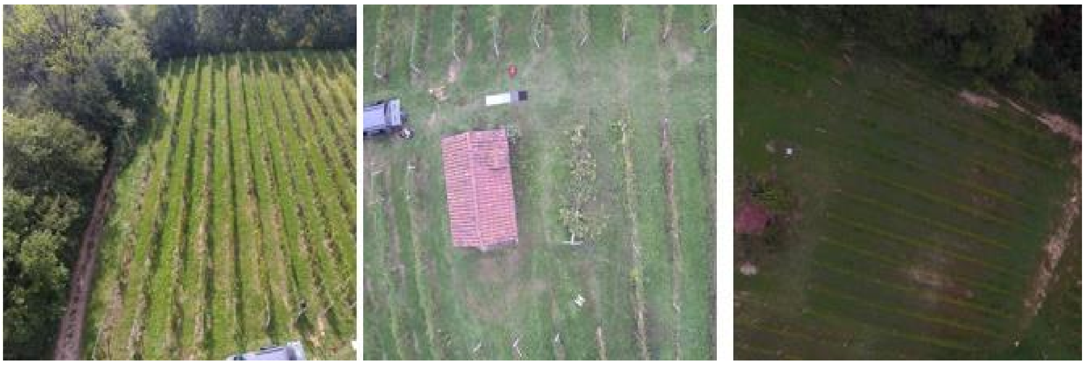


Figure 70. Original images: the result of acquisition configurations.

Thus, the annotation process was performed, assuring that all produced images contain more than one diseased area, and each diseased area has the same size. The labelling procedure was particularly challenging for the lack of shapes; indeed, the leaves shape is not evident at this scale. Thus, the size of a diseased area was digitized to cover the small diseased areas in a precise way. It is also evaluated the possibility to define bigger areas; however, the presence of sparse green leaves does not allow to use it. More in-depth, two classes are considered: FD diseased plants and background. The background class includes all the other objects that can be identified in the environment, such as healthy grapevine, terrain, poles, buildings, and other vegetation species. The annotation was carried out with an open-source software developed from MIT, LabelImage API [323], that allowed to identify the area and to label the classes manually.

As mentioned in Section 6.1, the binary classification task is applied to investigate the two different approaches: semantic segmentation and object detection. Thus, Random Forest and Faster R-CNN have been tested. As shown in the methodology workflow (Figure 68), the annotation step is different for the two algorithms due to the different expected outputs.

For what concerns the RF have been identified two classes: diseased area class (FD), and no diseased area class (NFD). In this case, the NFD is composed of all classes that it is possible to define for this environment (e.g., soil, grass, etc.) (Figure 71). As results of the annotation process, 2360 polygons have been labelled and stored in a .json file. Since the RF is used as pixel-wise techniques, 7000 pixels have been randomly chosen for the training and test of the algorithms.

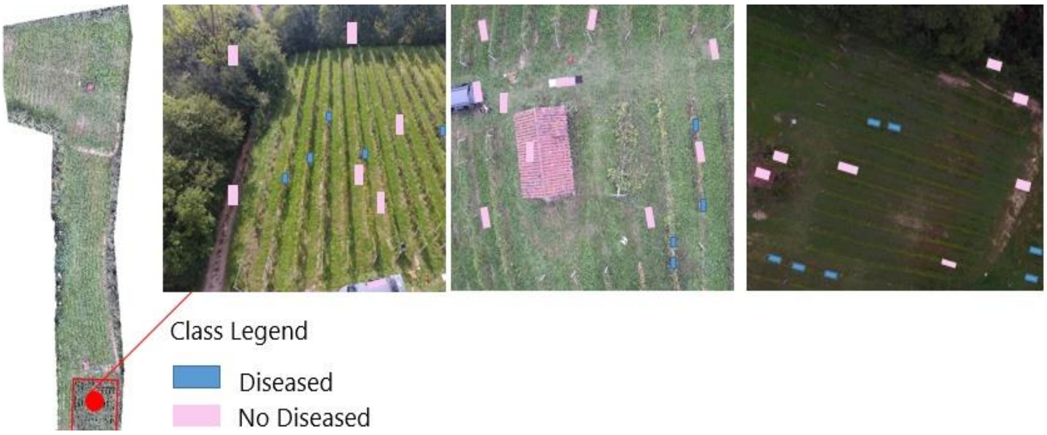


Figure 71. Random Forest: Data annotation. The diseased vineyard areas (in blue), and the background (in pink).

For the Faster R-CNN algorithm instead, as an object-detection technique, the FD class was the single possible class to annotate (Figure 72). Therefore, 4575 polygons were annotated and saved in TFRecord format. This format can handle the storage of a binary record sequence and allows efficient data reading [324].

Usually, the detection task is addressed with images in which objects cover a significant part of the image. In this case, instead, as shown in and Figure 72, the background prevailing on the diseased area, thus the detection is more challenging.

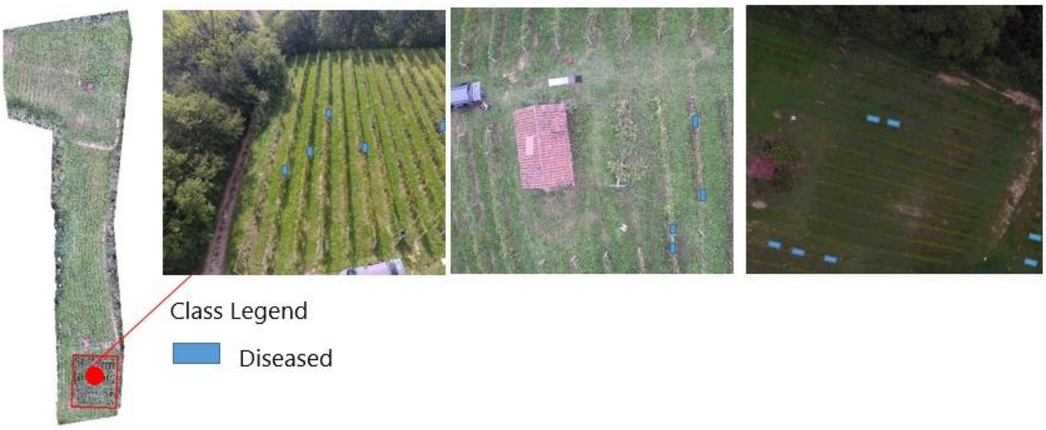


Figure 72. Faster R-CNN: Data annotation. The diseased areas of the vineyard (blue box).

After the dataset annotation, it was possible to randomly split the images into two sets: training (156 images) and test (76 images) for proceeding with the hyperparameter tuning of the two algorithms and the disease detection (Section 6.3). It worth to underline that the images were consistently distributed in the training and validation set based on an equilibrated number of different types of images (i.e., different scales, resolution). Moreover, both RF and Faster R-CNN used the same images for training and testing. In details, for the Random Forest, 5000-pixel samples are randomly selected from different images for the training and 2000-pixel samples for the testing. Furthermore, because of the imbalance between FD disease class and the background, a class balance rectification was made. The balance rectification consists of the setting of the same number of pixels for each class. To reach this purpose, a threshold has been fixed. For the Faster R-CNN, 3660 and 915 polygons are chosen for training and testing the model, respectively. Table 28 summarizes the main features of the annotation process and the training and test datasets prepared for the two algorithms.

Table 28. The results of the annotation process and the main specifications of the two datasets.

	Image N°	Dataset dimension	Training sample N°	Test sample N°	Total Annotation time
Random Forest	228	1.58 GB	5000 (pixels)	2000 (pixels)	48h
Faster R-CNN	228	1.58 GB	3660 (polygons)	915 (polygons)	48h

6.3 Hyperparameter tuning and disease detection

As reported in Chapter 5, also, in this case, the hyperparameter adjustment was performed on the training set. For what concerns the RF algorithm (Section 6.3.1), the number of features and the number of trees were tuned, fixing the maximum number of samples for each class. Instead, for the Faster R-CNN (Section 6.3.2), two backbone were tested. Moreover, the anchor boxes scale and ratio, the Intersection Over Union and training steps number ware changed.

The algorithms were implemented both in Python programing language. For the RF, the sklearn library was used [325], while for Faster R-CNN, a Tensorflow object detection API was employed [267]. Appendix D and Appendix E report the Python implementation of the two algorithms, RF and Faster R-CNN, respectively.

The out of bag score and testing accuracy have been evaluated to characterize the performance of Random Forest (section 4.2.1). For the Faster R-CNN, the average precision was used as evaluation metrics for Faster R-CNN (section 4.2.2). The tuning and the detection ware performed both for RF and Faster R-CNN on an Ubuntu workstation (18.04.4 LTS distribution) with an Intel(R) Xeon(R) CPU E5-1650 v4@3.60GHz (12 CPU with 6 cores per socket) and an NVIDIA GP102 - TitanX with 12 GB memory.

6.3.1 Random forest: hyperparameter tuning and detection results

The tuning of the hyperparameters for the Random Forest concerned the number of features and the number of estimators (trees) substantially. Based on the accuracy, the maximum number of trees possible in a forest has been set. The best result obtained corresponding to the number of trees equal to 21. For what concerns the features number, instead, it is worth to underline that we have only three features according to the channels of the sensors. The feature importance analysis has been demonstrated that all three features, which means the Red, Green and Blue have the same relevancy and thus are all essential for the detection. The training time was around10 hours.

The experimental analysis shows that an accuracy of 88% has been reached. The segmentation time was equal to 2 min/images. The out-of-bag analyses, on the other hand, underlines an encouraging possibility of generalization. Therefore, the test accuracy is 89% of the samples, considering all the different scenarios. Despite in the forest, some red areas related to possible diseased trees generate false positive in the analysis, as shown in Figure 73, it is possible to appreciate the graphical outcome and the robustness classification.

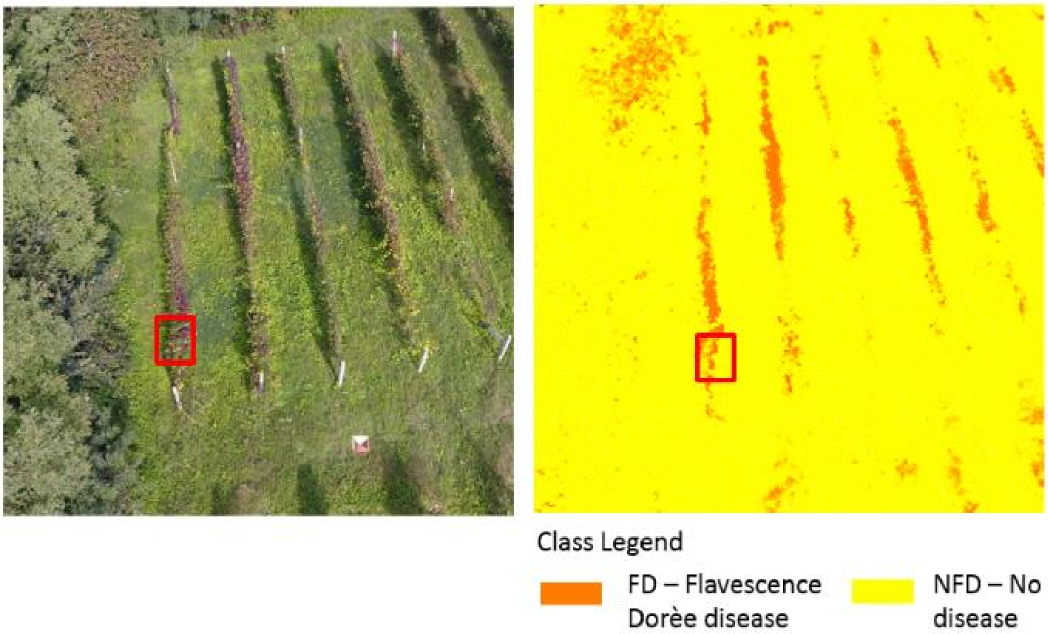


Figure 73. Random Forest test results. The original image (on the left), the classified image (on the right). In the red box shows a detected diseased area [269].

6.3.2 Faster R-CNN: hyperparameter tuning and detection results

As explained in section 4.2.2, the Faster R-CNN is composed of three different modules. In particular, different Faster R-CNN implementation can be composed of different types of backbones, such as ResNET or AlexNET. Moreover, according to the application complexity, the CNN backbone depth can be reduced. To select, the best implementation of the Faster R-CNN according to this application, the experiments were performed considering Faster R-CNN based on a backbone with two different depth: ResNet-50 (50-layers) and ResNet-101(101-layers) [326], trying to find a trade-off between accuracy and computational time. For a single-object detection (in our case the images contain as an object only the diseased plants), ResNet-50 could be enough. Thus, to test the performances of these algorithms in this PA application, the hyperparameter tuning was performed on each model. Pre-trained Faster R-CNN models with ImageNet weight has been selected and fine-tuned on the FD dataset, to decrease the computational time. However, all the Faster R-CNN modules, the backbone, the RPN and Fast R-CNN branch were trained in all the experiments. The momentum was set to 0.89, and the learning rate was set to 0.001, as suggested in the literature. Furthermore, based on the number of bounding boxes annotated per images (100 on average), a maximum number of proposals per class (FD class) was decreased from 300 to 100.

As mentioned above, the hyperparameters under investigation were the anchor box scales and ratios, the IoU, and the number of training steps. For anchor boxes, scale and ratio, two clusters of hyperparameters were selected. The first cluster has been chosen as suggested by the literature. Thus, [128^2,256^2,512^2] was identified for the anchor box scales and [0.5,1,2] for the ratio. For the second cluster, [0.25^2, 0.5^2, 1^2, 2^2] and [0.5,1,2] were set respectively for box scales and ratios considering the specific case application. The scale was set considering that the diseased leaves, can be classified as small objects. Instead, have still the same ratio. The value established for the Intersection over Union was 0.7 and 0.5. Even in this case, the literature value has been considered; however, higher is the value and higher is the discarded percentage. With the 0.7 the 50% of the boxes were not identified. Finally, the training steps were fixed based on the loss function of the training dataset and the validation accuracy. Table 29 reports the results of hyperparameter tuning.

Table 29. Faster R-CNN hyperparameter tuning. AP@50 is Average precision at 50% of Intersection over Union.

	Test	Anchor box (scale and ratios)	IoU	N of steps	AP@IoU50	Training time (h)	Test time (sec/step)
Faster R-CNN-ResNet-50	1	[128^2,256^2,512^2] [0.5, 1, 2]	0.7	60K	20%	14	0.3
	2	[0.25^2, 0.5^2, 1^2, 2^2] [0.5,1,2]	0.5	80K	65%	14	
Faster R-CNN- ResNet-101	3	[128^2,256^2,512^2] [0.5, 1, 2]	0.7	60K	40%	20	0.45
	4	[0.25^2, 0.5^2, 1^2, 2^2] ratio: [0.5,1,2]	0.5	80K	82%	20	

As Table 29 shows, for both tested architectures (Faster R-CNN_ResNet 50 and Faster R-CNN_ResNet 101) the smaller is the anchor boxes scale, the higher is the average precision. This means that the more the proposal and ground truth area are similar, the more the model can learn the real size of objects. Moreover, the average precision increase implies a more significant number of detected instances and the associated confidence score. For what concerns the testing time, Faster R-CNN_ResNet 101 takes more time than the Faster R-CNN_ResNet 50 for the training, with a speed of 0.45 sec/step instead of 0.3 sec/step.

The experimental analysis shows that a maximum average precision of is 82% is reached with Faster R-CNN_ResNet 101. The confidence score of the detected instances is greater than 80% in most of the cases. Figure 74 and Figure 75 present some results for Faster R-CNN_ResNet 101. As it is possible to notice, the model also reached a great generalization because diseased areas detection works with all types of images.

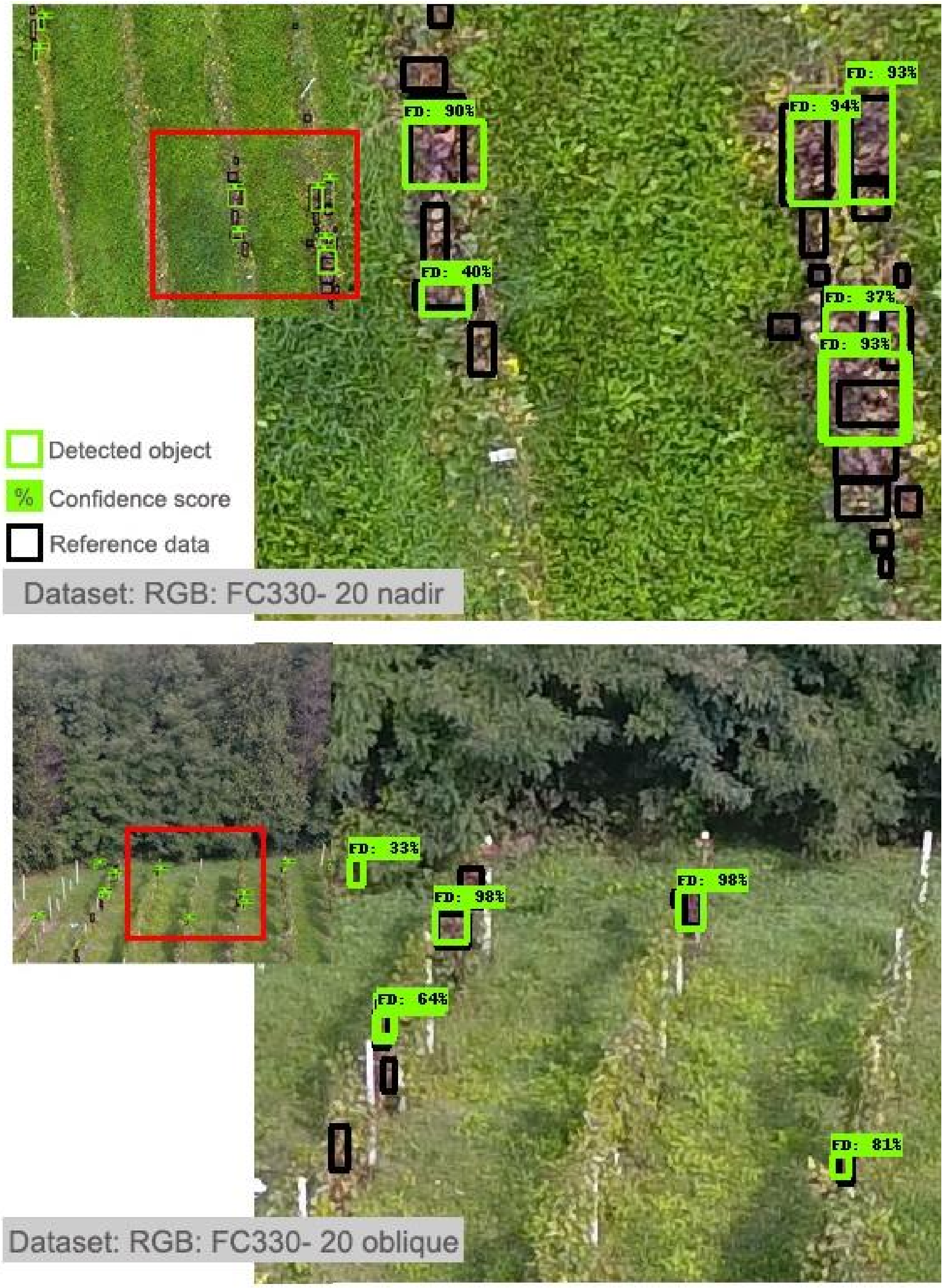


Figure 74. Faster R-CNN_ResNet 101 (test 4). Results of different types of images. In black, the reference data and in green detected diseased area.

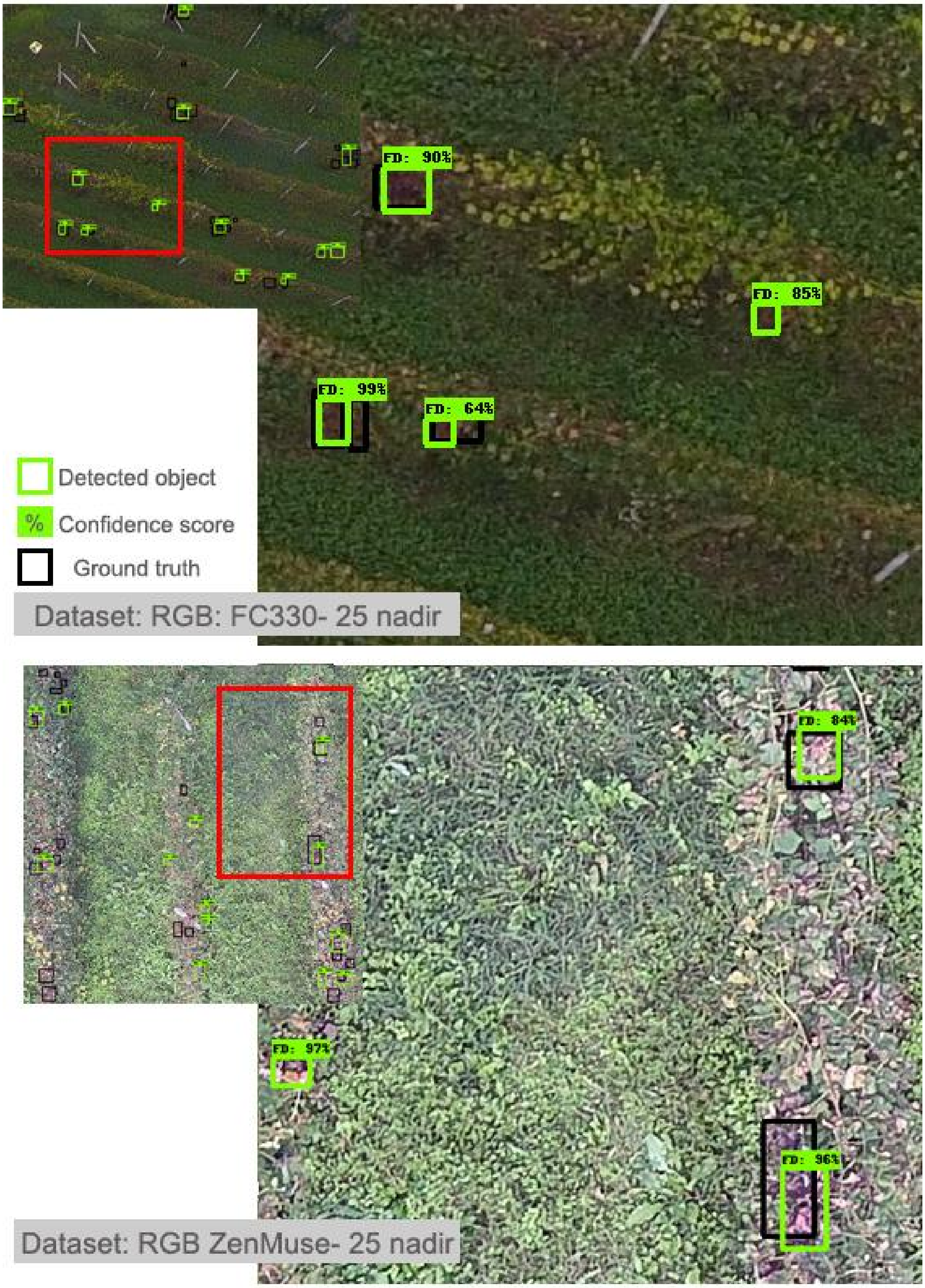


Figure 75. Faster R-CNN_ResNet 101 (test 4). Results of different types of images. In black, the reference data and in green detected diseased area.

6.4 Discussion

The application shows that for the two algorithms, the disease plant's detection is accomplished with high accuracy of 88% for the RF and a high average precision value of 82% for the Faster R-CNN_ResNet 101. Moreover, both the algorithms show a great model generalization. The RF presents an OOB score of 89%, while for the Faster R-CNN_ResNet 101, the model generalization can be deduced from Figure 74 and Figure 75. These figures, indeed, point out that the detection is accomplished in all the types of images.

These results can be considered outstanding, considering the type of dataset used for the detection is complex. Indeed, the detection, in this case, had to deal with:

(1) a dataset collected in the various conditions different illumination conditions, multiple-time, scales, and prospective);
(2) very small object, the diseased plant. It is worth to underline that even if the high spatial resolution (1 cm/ pixel in the worst case), it could be challenging to identify the leaves or plant shape at the canopy scale. Thus, the detection is more complicated. As reported in section 6.1 usually, laboratory images are employed for this detection task, with the well-defined condition and objects on a big scale (leaf scale);
(3) the similarity between plants and background.

The evaluation metrics for the two techniques are different; therefore, it is not possible considering a quantitative comparative analysis. However, it is possible to summarize some qualitative considerations.

The two algorithm presents a different annotation process. For the RF, all the classes in the image have to be labelled. Instead, for the Faster-RCNN, only the FD diseased class was annotated. Thus, the annotation process of a semantic segmentation method requires more effort.

Moreover, Figure 73 and Figure 74 display the difference in the visual results. The semantic segmentation outcome (Figure 73) is a thematic map which means that the whole images are classified in FD class and background. On the other hands, the object detection (Figure 74) localizes the diseased spots. Since the disease is localized in precise spots and on the plants, for avoiding the classification of a worthless part of the image, in most common cases, object detection can fit better to the task. Thus, it can be deduced that the Faster R-CNN_ResNet 101 can be more suitable for this application.

Chapter 7

Conclusion and perspectives

The present research work aims to develop a quasi real-time automatic information extraction technique applied to multi-sensory, multi-prospective, and multi-resolution data for monitoring and inspection. Two applications, considered as representative examples, have been analyzed: ice classification and plant disease detection in precision agriculture. This work provides a rapid and cost-effective strategy for remote object detection in both application fields. For the ice identification on aircrafts, a rapid detection system for de-icing allows to safeguard the lives of operators and passengers and to reduce air and water pollution. On the other hand, the detection of diseased plants improves the quality of the product and reduces the waste of pollutants. Therefore, as Chapter 1 pointed out, the impacts of the present research on the regional and urban management are significant in the framework of Sustainable Development Goals promoted by the United Nations. Moreover, the results are in line with the sustainability and technological innovation objectives promoted by the Interdepartmental Centre PIC4SeR, where this research has been carried out.

Despite the common purpose that endorses the efficiency improvement of both de-icing and precision viticulture processes, each application includes specific targets. Therefore, the remote system selection, whether concerns platform(s) or sensor(s) and the followed algorithm definition for the extraction of information by images, is strictly application dependent. Nevertheless, these applications serve as examples for adopting machine learning techniques to solve industrial and non-industrial environments.

Regarding the ice detection, the semantic segmentation and the exploiting of hyperspectral data were investigated. Instead, considering the precision agriculture application, the semantic segmentation and the object detection were tackled through RGB, multi-perspective, multi-sensory, and multi-resolution data. The same workflow was adopted for both the two applications: (1) the platform definition, (2) the choice of the sensor(s), (3) the algorithm selection for automatic information extraction, and (4) the performance analysis. The following sections present the key findings for both applications.

7.1 System configuration and calibration

For what concerns the system configuration, the definition of the platform-sensor system for the ice detection application was the most challenging aspect, as it requires the hyperspectral sensor. Indeed, the state of the art highlighted that light-weight UAV hyperspectral cameras, despite a better resolution compared with the RGB and multispectral camera, have not yet been well-developed and integrated technologies.

An in-depth analysis of the hyperspectral sensors on the market showed that the Senop Rikola camera was the best solution for our application. The Senop Rikola, was selected and integrated on a DJI Matrice 200, considering all the complementary hardware components. To manage and to process hyperspectral data in real-time from a GCS, a computer system and the transmission module were investigated and integrated onboard. Moreover, a gimbal was designed ad hoc for mounting the camera. However, the system requires several tests on the field for the system and sensor security. Thus, it is still under construction.

All the issues arisen in previous research related to the hyperspectral camera characterization were faced in the present work. The geometric calibration of the camera was performed developing a MATLAB routine, and the sensors were characterized by a geometric and radiometric point of view. Moreover, a routine for band-coregistration was developed and tested. In this way, the data collected with the Senop Rikola, geometrically and radiometrically corrected, can be used for the detection task.

7.2 Semantic segmentation using hyperspectral and multispectral images

As mentioned above and deeply explained in Chapter 5, the UAS is still under construction. Thus, to avoid the problem of not using it, for the hyperspectral data collection a terrestrial platform, equipped with all the components, was employed. This attempts at simulating the real detection of the ice for aircraft de-icing inspections. In this way, the transfer of this solution to a drone application has been guaranteed. Moreover, to simulate the case study more realistically, different types of ice were created in a laboratory.

Once the definitive system was set, and the ice samples were generated, the research dealt with the algorithm definition task, analyzing all the possible automatic information extraction techniques. Regarding algorithms, the state of the art highlighted that semantic segmentation, based on pixel-wise classification, is the most common approach for analyzing hyperspectral data. Therefore, Random forest (RF) and Support Vector Machine were chosen, and a comparative analysis was performed. The results, presented Chapter 5, underlined that SVM provided better accuracy while the faster classifier is the RF.

According to the general objective of a real-time process, the computation time was a crucial aspect evaluated. Even if the RF revealed to have the best processing time performance, it was not faster enough for real-time application. The high data storage computational efforts, indeed, affect the processing time. To allow the use of such a complex sensor output in a real-time process, two possible solutions to deal with the data type, and thus reducing the dimensionality, were proposed:

(1) the principal component extraction, applying a data transformation;

(2) the identification of significative original bands.

In both cases, the principal component analysis was used, and the impact of the dimensionality reduction was investigated in term of semantic segmentation accuracy.

The PCA results reveal the possibility of reducing the bands from 100 to at least 27 bands, with an improvement in computational time. However, due to the data transformation process, PCA does not allow to retrieve original bands. The knowledge of the significative subset of original bands can reduce processing time both for acquisition and analysis phase. It is evident that acquiring 100 bands requires more time than acquiring 27. Furthermore, the analysis of significant bands highlights the possibility not only to speed up the use of the hyperspectral camera but also to substitute the hyperspectral sensor with an equivalent multispectral. In this way, the system becomes more accessible in terms of cost and, also in terms of data management. (Chapter 5). Although the results related to the multispectral camera appear not to satisfy the application requirements for what concerns the accuracy, the improvements in terms of the computational time were significant. Furthermore, the outcomes of this thesis show the possibility to select an ad hoc multispectral camera, effective both in terms of computational time and accuracy for ice detection. In the end, two datasets were considered (acquired under different conditions and resolution). The PCA results on the two datasets demonstrate that the significative bands are recurrent over time. This is a promising outcome because it is possible to assume the transferability of the results on new datasets. This means that the results remain satisfactory, also reducing the hypercube as in previous experiments. This outcome also has great prospective for the detection because it allows, thanks to the band stability, to transfer the training. Despite this transfer learning requires further investigations, it represents a remarkable improvement in reducing the processing time.

7.3 From semantic segmentation to object detection using RGB images

Conversely to the ice detection, the plant disease detection required a drone survey, and thus it was not possible to simulate real condition in the laboratory using ground platforms. Furthermore, the state of the art highlighted the need to study these problems in real conditions. The study of plants requires several points of view and an accurate degree of detail. It is easier to recognize the diseased leaf with a close-range survey than with aerial views, which, on the other hand, allow an overall view. RGB sensors, already well-integrated in UAV, replaced the hyperspectral camera. The feasibility to generalize an algorithm for detection was investigated.

For this purpose, two sensors, with different flight heights and patterns, were employed for data collection, exploiting not only the possibility to see plants from the nadiral perspective (as usually reported in the literature) but also from a side view with multiple resolutions. As shown in Chapter 6, this has a great impact on the interpretation. After data acquisition, the automatic extraction algorithm was defined. The need to detect only diseased plants (binary classification), and not to segment the entire image, was translated into the shift from a pixel-wise algorithm to an object-based. From an application point of view, segmenting the entire scene was not very practical.

An analysis of RF (semantic segmentation algorithm) and Faster R-CNN (object detection algorithm) was carried out to demonstrate this statement. The RF was employed according to the results of Chapter 5, and its effectiveness in computational terms was demonstrated. The Faster R-CNN was select in according to the analysis of state of the art. The results obtained have demonstrated promising benefits (88% of accuracy) in disease detection and a suitable generalization of the Random Forest model. For what concerns the Faster R-CNN, an excellent accuracy in results (82%) were shown, demonstrating competitive in processing time compared to the RF. A suitable generalization of the model, even in this case, was obtained. Indeed, the diseased plants were recognized in all the proposed images with a confidence score of 80%. The generalization of the model, using multi-sensory, multi-perspective, and multi-resolution data is an outstanding result and can be a starting point for further analysis.

7.4 Summary of key findings and future perspective

The results obtained in the two applications highlight the following key findings. Firstly, the hyperspectral sensor provides a profitable solution in discriminating non-visible objects such as ice, difficult to detect with RGB sensors. Then, the discriminating band identification leads to data management benefits and provides a less expensive and more suitable solution, even adopting multispectral sensors. Secondly, object detection real-time algorithms, as the Faster R-CNN, reveal to be very effective for detecting single-class detection, and, if well-generalized, it allows to transfer of the learning from a dataset to another. All ML algorithms used in this work are developed in the open-source Python language, and thus the integration in SDK of drones and software, compatible with this language, can be feasible.

Although this research work results are promising and offer enhanced real-time detection UAV application methods, some future developments can be drawn.

Regarding the hyperspectral camera-drone integration, the embedded computer module of the UAV could be programmed to manage also the hyperspectral camera, avoiding the use of an additional computing module and reducing the payload weight. Further analyses can be developed for the georeferencing of hyperspectral imaging without GCPs.

In the present thesis, the use of hyperspectral data was limited to the case of the ice detection, considered as representative for other cases, and it could also be extended to precision agriculture application. For example, in plant disease detection, hyperspectral data could provide an added value for early plant disease detection.

For what concerns, instead, the detection tasks, further research could explore the algorithms adopted in this thesis, customized for two very specific application field, expanding their use in other industrial and non-industrial application fields, such as the detection of collapsed buildings after a natural disaster, or for other precision agriculture tasks.

In details, the investigation of hyperspectral, multi-perspective, and multi-resolution data combined with a deep learning algorithm, such as the Faster RCNN, could allow to manage complex data with a potential, versatile and speed algorithm. Moreover, it would be useful to continue training the same algorithms with different data to increase its ability to discriminate more details and multiple objects. At the same time, the possibility to improve the capability of transfer learning could be investigated, providing an enhancement in detection.

References

[1] M. Dunbabin and L. Marques, ‘Robots for Environmental Monitoring: Significant Advancements and Applications’, *IEEE Robotics Automation Magazine*, vol. 19, no. 1, pp. 24–39, Mar. 2012, doi: 10.1109/MRA.2011.2181683.

[2] X. He *et al.*, ‘Autonomous Chemical-Sensing Aerial Robot for Urban/Suburban Environmental Monitoring’, *IEEE Systems Journal*, vol. 13, no. 3, pp. 3524–3535, Sep. 2019, doi: 10.1109/JSYST.2019.2905807.

[3] M. Trincavelli, M. Reggente, S. Coradeschi, A. Loutfi, H. Ishida, and A. J. Lilienthal, ‘Towards environmental monitoring with mobile robots’, in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nice, Sep. 2008, pp. 2210–2215, doi: 10.1109/IROS.2008.4650755.

[4] A. H. While, S. Marvin, and M. Kovacic, ‘Urban robotic experimentation: San Francisco, Tokyo and Dubai’, *Urban Studies*, p. 0042098020917790, Jun. 2020, doi: 10.1177/0042098020917790.

[5] G. Quaglia, P. Cavallone, and C. Visconte, ‘Agri_q: Agriculture UGV for Monitoring and Drone Landing’, in *Mechanism Design for Robotics*, vol. 66, A. Gasparetto and M. Ceccarelli, Eds. Cham: Springer International Publishing, 2019, pp. 413–423.

[6] K. R. Aravind, P. Raja, and M. Pérez-Ruiz, ‘Task-based agricultural mobile robots in arable farming: A review’, *Span J Agric Res*, vol. 15, no. 1, p. e02R01, Apr. 2017, doi: 10.5424/sjar/2017151-9573.

[7] G. Tmušić *et al.*, ‘Current Practices in UAS-based Environmental Monitoring’, *Remote Sensing*, vol. 12, no. 6, p. 1001, Mar. 2020, doi: 10.3390/rs12061001.

[8] M. F. Othman and K. Shazali, ‘Wireless Sensor Network Applications: A Study in Environment Monitoring System’, *Procedia Engineering*, vol. 41, pp. 1204–1210, 2012, doi: 10.1016/j.proeng.2012.07.302.

[9] Y. Liu, J. Huang, and N. Lu, ‘Research on Environmental Monitoring System Based on Microservices and Data Mining’, *E3S Web Conf.*, vol. 143, p. 02031, 2020, doi: 10.1051/e3sconf/202014302031.

[10] A. Ballesteros-Gómez and S. Rubio, ‘Recent Advances in Environmental Analysis’, *Anal. Chem.*, vol. 83, no. 12, pp. 4579–4613, Jun. 2011, doi: 10.1021/ac200921j.

[11] M.-F. R. Lee, F. H. S. Chiu, and C. Zhuo, ‘Remote sensing and analysis using autonomous mobile robot with onboard micro-spectrometer’, in *2013 International Conference on Fuzzy Theory and Its Applications (iFUZZY)*, Taipei, Taiwan, Dec. 2013, pp. 195–200, doi: 10.1109/iFuzzy.2013.6825435.

[12] M. Hutchinson, H. Oh, and W.-H. Chen, ‘A review of source term estimation methods for atmospheric dispersion events using static or mobile

sensors', *Information Fusion*, vol. 36, pp. 130–148, Jul. 2017, doi: 10.1016/j.inffus.2016.11.010.

[13] D. Belanche, L. V. Casaló, C. Flavián, and J. Schepers, 'Service robot implementation: a theoretical framework and research agenda', *The Service Industries Journal*, vol. 40, no. 3–4, pp. 203–225, Mar. 2020, doi: 10.1080/02642069.2019.1672666.

[14] Intl. Federation of Robotics (IFR), 'International Federation of Robotics'. https://ifr.org/service-robots (accessed Aug. 20, 2020).

[15] T. Haidegger *et al.*, 'Applied ontologies and standards for service robots', *Robotics and Autonomous Systems*, vol. 61, no. 11, pp. 1215–1223, Nov. 2013, doi: 10.1016/j.robot.2013.05.008.

[16] ISO, 'ISO 8373:2012(en), Robots and robotic devices — Vocabulary', 2012. https://www.iso.org/obp/ui/#iso:std:iso:8373:ed-2:v1:en:term:2.10 (accessed Aug. 19, 2020).

[17] S. García, D. Strüber, D. Brugali, T. Berger, and P. Pelliccione, 'Robotics Software Engineering: A Perspective from the Service Robotics Domain', *arXiv:2006.10608 [cs]*, Jul. 2020, doi: 10.1145/3368089.3409743.

[18] J. Wirtz *et al.*, 'Brave new world: service robots in the frontline', *Journal of Service Management*, vol. 29, no. 5, pp. 907–931, Oct. 2018, doi: 10.1108/JOSM-04-2018-0119.

[19] K. Scharei, F. Heidecker, and M. Bieshaar, 'Knowledge Representations in Technical Systems -- A Taxonomy', *arXiv:2001.04835 [cs]*, Jan. 2020, Accessed: Aug. 20, 2020. [Online]. Available: http://arxiv.org/abs/2001.04835.

[20] van Est, Q.C., Bunders, D., Korthagen, I., and Technology, Innovation & Society, 'Rise of robot city politics', 2017, Accessed: Aug. 23, 2020. [Online]. Available: https://research.tue.nl/nl/publications/rise-of-robot-city-politics(e70b1dc4-fbd2-4b45-98f7-881d6783592b).html.

[21] J. Andreu-Perez, F. Deligianni, D. Ravi, and G.-Z. Yang, 'Artificial Intelligence and Robotics', *arXiv:1803.10813 [cs]*, Mar. 2018, Accessed: Aug. 20, 2020. [Online]. Available: http://arxiv.org/abs/1803.10813.

[22] M.-H. Huang and R. T. Rust, 'Artificial Intelligence in Service', *Journal of Service Research*, vol. 21, no. 2, pp. 155–172, May 2018, doi: 10.1177/1094670517752459.

[23] K. Rajan and A. Saffiotti, 'Towards a science of integrated AI and Robotics', *Artificial Intelligence*, vol. 247, pp. 1–9, Jun. 2017, doi: 10.1016/j.artint.2017.03.003.

[24] Alexandra Garr-Schultz and Wendi Gardner, 'Strategic Self-Presentation of Women in STEM', *Social Sciences*, vol. 7, no. 2, p. 20, Jan. 2018, doi: 10.3390/socsci7020020.

[25] M. Ehlers, M. Gähler, and R. Janowsky, 'Automated analysis of ultra high resolution remote sensing data for biotope type mapping: new possibilities and challenges', *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 57, no. 5–6, pp. 315–326, Apr. 2003, doi: 10.1016/S0924-2716(02)00161-2.

[26] I. Tiddi, E. Bastianelli, E. Daga, M. d'Aquin, and E. Motta, 'Robot–City Interaction: Mapping the Research Landscape—A Survey of the Interactions Between Robots and Modern Cities', *Int J of Soc Robotics*, vol. 12, no. 2, pp. 299–324, May 2020, doi: 10.1007/s12369-019-00534-x.

[27] D. of E. and S. A. United Nations, 'THE 17 GOALS', *Sustainable Development*. https://sdgs.un.org/goals (accessed Aug. 14, 2020).

[28] United Nations (UN), ‘Sustainable Development Goals: 17 Goals to Transform our World’, in *Furthering the Work of the United Nations. Highlights of the Tenure of Secretary-General Ban Ki-moon 2007-2016*, United Nation Publications., New York, NY, 2016, p. 349.

[29] J. Sachs and K. Pan, ‘The Age of Sustainable Development’, p. 565.

[30] P. Dauvergne, ‘The globalization of artificial intelligence: consequences for the politics of environmentalism’, *Globalizations*, pp. 1–15, Jun. 2020, doi: 10.1080/14747731.2020.1785670.

[31] R. MSc Tsvetkova, ‘What does Industry 4.0 mean for Sustainable Development?’, *INTERNATIONAL SCIENTIFIC JOURNAL ‘INDUSTRY 4.0’*, no. 6, pp. 294–297, 20017.

[32] R. Macrorie, S. Marvin, and A. While, ‘Robotics and automation in the city: a research agenda’, *Urban Geography*, pp. 1–21, Dec. 2019, doi: 10.1080/02723638.2019.1698868.

[33] C. Coletta, L. Heaphy, and R. Kitchin, ‘From the accidental to articulated smart city: The creation and work of “Smart Dublin”’, Mar. 09, 2017.

[34] M. Raad, ‘A New Business Intelligence Emerges: Geo.AI’, *ESRI, Artificial Intelligence- AI*, p. 5, Apr. 2017.

[35] R. Rivera, M. Amorim, and J. Reis, ‘Robotic Services in Smart Cities: An Exploratory Literature Review’, in *2020 15th Iberian Conference on Information Systems and Technologies (CISTI)*, Sevilla, Spain, Jun. 2020, pp. 1–7, doi: 10.23919/CISTI49556.2020.9141099.

[36] L. Leydesdorff and M. Deakin, ‘The Triple-Helix Model of Smart Cities: A Neo-Evolutionary Perspective’, *Journal of Urban Technology*, vol. 18, pp. 53–63, Apr. 2011, doi: 10.1080/10630732.2011.601111.

[37] D. Maye, ‘“Smart food city”: Conceptual relations between smart city planning, urban food systems and innovation theory’, *City, Culture and Society*, vol. 16, pp. 18–24, Mar. 2019, doi: 10.1016/j.ccs.2017.12.001.

[38] O. Golubchikov and M. Thornbush, ‘Artificial Intelligence and Robotics in Smart City Strategies and Planned Smart Development’, *Smart Cities*, vol. 3, no. 4, pp. 1133–1144, Oct. 2020, doi: 10.3390/smartcities3040056.

[39] M. Nagenborg, ‘Urban robotics and responsible urban innovation’, *Ethics Inf Technol*, vol. 22, no. 4, pp. 345–355, Dec. 2020, doi: 10.1007/s10676-018-9446-8.

[40] United Nation, ‘The New Urban Agenda’, *Habitat III*, 2017. http://habitat3.org/the-new-urban-agenda/ (accessed Aug. 04, 2020).

[41] PIC4SeR, ‘PoliTO Interdepartmental Centre for Service Robotics’, 2017. https://pic4ser.polito.it/ (accessed Aug. 04, 2020).

[42] M. A. Musci, ‘Le attività del centroPIC4SeR@Polito: robotica di servizio per l’agricoltura di precisione’, Bolzano, Italy, 29- 11 2018, pp. 723–724, [Online]. Available: http://atti.asita.it/ASITA2018/Pdf/096.pdf.

[43] J. Jung, M. Maeda, A. Chang, M. Bhandari, A. Ashapure, and J. Landivar-Bowles, ‘The potential of remote sensing and artificial intelligence as tools to improve the resilience of agriculture production systems’, *Current Opinion in Biotechnology*, vol. 70, pp. 15–22, 2021, doi: https://doi.org/10.1016/j.copbio.2020.09.003.

[44] S. Salcedo-Sanz *et al.*, ‘Machine learning information fusion in Earth observation: A comprehensive review of methods, applications and data

sources', *Information Fusion*, vol. 63, pp. 256–272, Nov. 2020, doi: 10.1016/j.inffus.2020.07.004.

[45] B. Liu, Y. Li, C. Liu, F. Xie, and J.-P. Muller, 'Hyperspectral Features of Oil-Polluted Sea Ice and the Response to the Contamination Area Fraction', *Sensors*, vol. 18, no. 2, p. 234, Jan. 2018, doi: 10.3390/s18010234.

[46] P. Moghadam, D. Ward, E. Goan, S. Jayawardena, P. Sikka, and E. Hernandez, 'Plant Disease Detection Using Hyperspectral Imaging', in *2017 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, Sydney, NSW, Nov. 2017, pp. 1–8, doi: 10.1109/DICTA.2017.8227476.

[47] S. Prasad and J. Chanussot, Eds., *Hyperspectral Image Analysis: Advances in Machine Learning and Signal Processing*. Cham: Springer International Publishing, 2020.

[48] L. Breiman, 'Random Forests', *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct. 2001, doi: 10.1023/A:1010933404324.

[49] C. Cortes and V. Vapnik, 'Support-vector networks', *Mach Learn*, vol. 20, no. 3, pp. 273–297, Sep. 1995, doi: 10.1007/BF00994018.

[50] B. Benjdira, T. Khursheed, A. Koubaa, A. Ammar, and K. Ouni, 'Car Detection using Unmanned Aerial Vehicles: Comparison between Faster R-CNN and YOLOv3', *arXiv:1812.10968 [cs]*, Dec. 2018, Accessed: May 07, 2020. [Online]. Available: http://arxiv.org/abs/1812.10968.

[51] Y. Li, K. Fu, H. Sun, and X. Sun, 'An Aircraft Detection Framework Based on Reinforcement Learning and Convolutional Neural Networks in Remote Sensing Images', *Remote Sensing*, vol. 10, no. 2, Art. no. 2, Feb. 2018, doi: 10.3390/rs10020243.

[52] B. Zoph, E. D. Cubuk, G. Ghiasi, T.-Y. Lin, J. Shlens, and Q. V. Le, 'Learning Data Augmentation Strategies for Object Detection', *arXiv:1906.11172 [cs]*, Jun. 2019, Accessed: Nov. 25, 2020. [Online]. Available: http://arxiv.org/abs/1906.11172.

[53] U. Alganci, M. Soydas, and E. Sertel, 'Comparative Research on Deep Learning Approaches for Airplane Detection from Very High-Resolution Satellite Images', p. 28, 2020.

[54] S. Ren, K. He, R. Girshick, and J. Sun, 'Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks', *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017, doi: 10.1109/TPAMI.2016.2577031.

[55] C. J. Tomlinson, L. Chapman, J. E. Thornes, and C. Baker, 'Remote sensing land surface temperature for meteorology and climatology: a review', *Meteorological Applications*, vol. 18, no. 3, pp. 296–306, 2011, doi: https://doi.org/10.1002/met.287.

[56] B. Park and R. Lu, Eds., *Hyperspectral Imaging Technology in Food and Agriculture*. New York, NY: Springer New York, 2015.

[57] D. Manolakis, R. Lockwood, and T. Cooley, *Hyperspectral Imaging Remote Sensing: Physics, Sensors, and Algorithms*. Cambridge: Cambridge University Press, 2016.

[58] S. M. de Jong and F. D. van der Meer, Eds., *Remote sensing image analysis: including the spatial domain*. Dordrecht: Springer, 2004.

[59] M. T. Melis and M. A. Dessena, *Il telerilevamento applicato*. 2006.

[60] Humboldt State University, ‘GSP 216. Introduction to Remote Sensing’, *Humboldt State Geospatial Online*, 2019. http://gsp.humboldt.edu/OLM/Courses/GSP_216_Online/lesson2-1/index.html (accessed Nov. 23, 2020).

[61] K. Tempfli *et al.*, *Principles of remote sensing: an introductory textbook.*, The International Institute for Geo-Information Science and Earth Observation(ITC). Enschede: ITC, 2009.

[62] C. Toth and G. Jóźków, ‘Remote sensing platforms and sensors: A survey’, *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 115, pp. 22–36, May 2016, doi: 10.1016/j.isprsjprs.2015.10.004.

[63] C. Zhang and J. M. Kovacs, ‘The application of small unmanned aerial systems for precision agriculture: a review’, *Precision Agric*, vol. 13, no. 6, pp. 693–712, Dec. 2012, doi: 10.1007/s11119-012-9274-5.

[64] J. Fussell and D. Rundquist, ‘On Defining Remote Sensing’, *Photogrammetric Engineering*, p. 5, 1986.

[65] D. Deery, J. Jimenez-Berni, H. Jones, X. Sirault, and R. Furbank, ‘Proximal Remote Sensing Buggies and Potential Applications for Field-Based Phenotyping’, *Agronomy*, vol. 4, no. 3, Art. no. 3, Sep. 2014, doi: 10.3390/agronomy4030349.

[66] J. R. Jensen, *Remote sensing of the environment: an earth resource perspective*, 2nd edition. Harlow: Pearson, 2014.

[67] H. Yao, R. Qin, and X. Chen, ‘Unmanned Aerial Vehicle for Remote Sensing Applications—A Review’, *Remote Sensing*, vol. 11, no. 12, p. 1443, Jun. 2019, doi: 10.3390/rs11121443.

[68] K. Whitehead and C. H. Hugenholtz, ‘Remote sensing of the environment with small unmanned aircraft systems (UASs), part 1: a review of progress and challenges’, vol. 2, p. 18, 2014.

[69] L. Tang and G. Shao, ‘Drone remote sensing for forestry research and practices’, *J. For. Res.*, vol. 26, no. 4, pp. 791–797, Dec. 2015, doi: 10.1007/s11676-015-0088-y.

[70] I. Colomina and P. Molina, ‘Unmanned aerial systems for photogrammetry and remote sensing: A review’, *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 92, pp. 79–97, Jun. 2014, doi: 10.1016/j.isprsjprs.2014.02.013.

[71] L. Pádua *et al.*, ‘UAS, sensors, and data processing in agroforestry: a review towards practical applications’, *International Journal of Remote Sensing*, vol. 38, no. 8–10, pp. 2349–2391, May 2017, doi: 10.1080/01431161.2017.1297548.

[72] X. Zhang, L. Han, L. Han, and L. Zhu, ‘How Well Do Deep Learning-Based Methods for Land Cover Classification and Object Detection Perform on High Resolution Remote Sensing Imagery?’, *Remote Sensing*, vol. 12, no. 3, p. 417, Jan. 2020, doi: 10.3390/rs12030417.

[73] P. J. Hardin and R. R. Jensen, ‘Small-Scale Unmanned Aerial Vehicles in Environmental Remote Sensing: Challenges and Opportunities’, *GIScience & Remote Sensing*, vol. 48, no. 1, pp. 99–111, Jan. 2011, doi: 10.2747/1548-1603.48.1.99.

[74] F. Nex and F. Remondino, ‘UAV for 3D mapping applications: a review’, *Applied geomatics*, no. 6, pp. 1–15, 2014, doi: 10.1007/s12518-013-0120-x.

[75] A. C. Watts, V. G. Ambrosia, and E. A. Hinkley, ‘Unmanned Aircraft Systems in Remote Sensing and Scientific Research: Classification and

Considerations of Use', *Remote Sensing*, vol. 4, no. 6, pp. 1671–1692, Jun. 2012, doi: 10.3390/rs4061671.

[76] H. Shakhatreh *et al.*, 'Unmanned Aerial Vehicles (UAVs): A Survey on Civil Applications and Key Research Challenges', *IEEE Access*, vol. 7, pp. 48572–48634, 2019, doi: 10.1109/ACCESS.2019.2909530.

[77] A. G. Korchenko and O. S. Illyash, 'The generalized classification of Unmanned Air Vehicles', in *2013 IEEE 2nd International Conference Actual Problems of Unmanned Air Vehicles Developments Proceedings (APUAVD)*, Oct. 2013, pp. 28–34, doi: 10.1109/APUAVD.2013.6705275.

[78] K. Dalamagkidis, 'Classification of uavs', *Handbook of unmanned aerial vehicles*, pp. 83–91, 2015.

[79] P. van Blijenburgh, *2018- RPAS Yearbook. Remotely Piloted Aircraft Systems.The Global Perspective.*, 15th ed. Paris, France: Blyenburgh & CO, 2018.

[80] P. S. Thenkabail, J. G. Lyon, and A. Huete, Eds., *Fundamentals, Sensor Systems, Spectral Libraries, and Data Mining for Vegetation: Hyperspectral Remote Sensing of Vegetation*, 2nd ed. CRC Press, 2018.

[81] P. K. R. Maddikunta *et al.*, 'Unmanned Aerial Vehicles in Smart Agriculture: Applications, Requirements and Challenges', *arXiv:2007.12874 [cs, eess]*, Jul. 2020, Accessed: Oct. 21, 2020. [Online]. Available: http://arxiv.org/abs/2007.12874.

[82] B. Hament and P. Oh, 'Unmanned aerial and ground vehicle (UAV-UGV) system prototype for civil infrastructure missions', in *2018 IEEE International Conference on Consumer Electronics (ICCE)*, Las Vegas, NV, Jan. 2018, pp. 1–4, doi: 10.1109/ICCE.2018.8326346.

[83] M. A. Gomarasca, *Basics of geomatics*. Dordrecht ; London: Springer, 2009.

[84] S. Liang, *Quantitative Remote Sensing of Land Surfaces*. United States ot America: Wiley - Interscience, 2004.

[85] 'Principles of Remote Sensing - Centre for Remote Imaging, Sensing and Processing, CRISP'. https://crisp.nus.edu.sg/~research/tutorial/optical.htm (accessed Jan. 01, 2021).

[86] B. Stark, M. McGee, and Y. Chen, 'Short wave infrared (SWIR) imaging systems using small Unmanned Aerial Systems (sUAS)', in *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*, Jun. 2015, pp. 495–501, doi: 10.1109/ICUAS.2015.7152328.

[87] A. Jenal, G. Bareth, A. Bolten, C. Kneer, I. Weber, and J. Bongartz, 'Development of a VNIR/SWIR Multispectral Imaging System for Vegetation Monitoring with Unmanned Aerial Vehicles', *Sensors*, vol. 19, no. 24, Art. no. 24, Jan. 2019, doi: 10.3390/s19245507.

[88] T. Adão *et al.*, 'Hyperspectral Imaging: A Review on UAV-Based Sensors, Data Processing and Applications for Agriculture and Forestry', *Remote Sensing*, vol. 9, no. 11, p. 1110, Nov. 2017, doi: 10.3390/rs9111110.

[89] J. M. Palmer and B. G. Grant, *The art of radiometry*. Bellingham, Wash: SPIE Press, 2010.

[90] O. Burggraaff *et al.*, 'Standardized spectral and radiometric calibration of consumer cameras', *Opt. Express, OE*, vol. 27, no. 14, pp. 19075–19101, Jul. 2019, doi: 10.1364/OE.27.019075.

[91] 'Hasselblad'. https://www.hasselblad.com/ (accessed Jan. 02, 2021).

[92] R.-M. Coliban, M. Marincaş, C. Hatfaludi, and M. Ivanovici, ‘Linear and Non-Linear Models for Remotely-Sensed Hyperspectral Image Visualization’, *Remote Sensing*, vol. 12, no. 15, Art. no. 15, Jan. 2020, doi: 10.3390/rs12152479.

[93] J. O’Connor and M. Smith, ‘UAV Camera Review for Surveying’, *Geo-matching*. https://geo-matching.com/content/uav-camera-review-for-surveying (accessed Oct. 22, 2020).

[94] M. H. Yun, J. Kim, D. Seo, J. Lee, and C. Choi, ‘Application possibility of smartphone as payload for photogrammetric UAV system’, *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*, vol. XXXIX-B4, pp. 349–352, Jul. 2012, doi: 10.5194/isprsarchives-XXXIX-B4-349-2012.

[95] J. Kim, S. Lee, H. Ahn, D. Seo, S. Park, and C. Choi, ‘Feasibility of employing a smartphone as the payload in a photogrammetric UAV system’, *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 79, pp. 1–18, May 2013, doi: 10.1016/j.isprsjprs.2013.02.001.

[96] L. Zhu, J. Suomalainen, J. Liu, J. Hyyppä, H. Kaartinen, and H. Haggren, ‘A Review: Remote Sensing Sensors’, in *Multi-purposeful Application of Geospatial Data*, R. B. Rustamov, S. Hasanova, and M. H. Zeynalova, Eds. InTech, 2018.

[97] S. Nebiker, N. Lack, M. Abächerli, and S. Läderach, ‘Light-weight multispectral UAV sensors and their capabilities for predicting grain yield and detecting plant diseases’, *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*, vol. XLI-B1, pp. 963–970, Jun. 2016, doi: 10.5194/isprsarchives-XLI-B1-963-2016.

[98] D. Zhao, L. Huang, J. Li, and J. Qi, ‘A comparative analysis of broadband and narrowband derived vegetation indices in predicting LAI and CCD of a cotton canopy’, *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 62, no. 1, pp. 25–33, May 2007, doi: 10.1016/j.isprsjprs.2007.01.003.

[99] P. H. Sampson, P. J. Zarco-Tejada, G. H. Mohammed, J. R. Miller, and T. L. Noland, ‘Hyperspectral Remote Sensing of Forest Condition: Estimating Chlorophyll Content in Tolerant Hardwoods’, *for sci*, vol. 49, no. 3, pp. 381–391, Jun. 2003, doi: 10.1093/forestscience/49.3.381.

[100] J. M. Fernández-Guisuraga, E. Sanz-Ablanedo, S. Suárez-Seoane, and L. Calvo, ‘Using Unmanned Aerial Vehicles in Postfire Vegetation Survey Campaigns through Large and Heterogeneous Areas: Opportunities and Challenges’, *Sensors*, vol. 18, no. 2, Art. no. 2, Feb. 2018, doi: 10.3390/s18020586.

[101] L. Deng, Z. Mao, X. Li, Z. Hu, F. Duan, and Y. Yan, ‘UAV-based multispectral remote sensing for precision agriculture: A comparison between different cameras’, *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 146, pp. 124–136, Dec. 2018, doi: 10.1016/j.isprsjprs.2018.09.008.

[102] H. Lu, T. Fan, P. Ghimire, and L. Deng, ‘Experimental Evaluation and Consistency Comparison of UAV Multispectral Minisensors’, *Remote Sensing*, vol. 12, no. 16, Art. no. 16, Jan. 2020, doi: 10.3390/rs12162542.

[103] S. Manfreda *et al.*, ‘On the Use of Unmanned Aerial Systems for Environmental Monitoring’, *Remote Sensing*, vol. 10, no. 4, Art. no. 4, Apr. 2018, doi: 10.3390/rs10040641.

[104] E. Adamopoulos and F. Rinaudo, ‘UAS-Based Archaeological Remote Sensing: Review, Meta-Analysis and State-of-the-Art’, *Drones*, vol. 4, no. 3, Art. no. 3, Sep. 2020, doi: 10.3390/drones4030046.

[105] *Hyperspectral Imaging for Food Quality Analysis and Control*. Elsevier, 2010.

[106] A. Zia, J. Liang, J. Zhou, and Y. Gao, ‘3D Reconstruction from Hyperspectral Images’, in *2015 IEEE Winter Conference on Applications of Computer Vision*, Waikoloa, HI, USA, Jan. 2015, pp. 318–325, doi: 10.1109/WACV.2015.49.

[107] Y. Zhong *et al.*, ‘Mini-UAV-Borne Hyperspectral Remote Sensing: From Observation and Processing to Applications’, *IEEE Geosci. Remote Sens. Mag.*, vol. 6, no. 4, pp. 46–62, Dec. 2018, doi: 10.1109/MGRS.2018.2867592.

[108] R. A. de Oliveira, A. M. G. Tommaselli, and E. Honkavaara, ‘Geometric Calibration of a Hyperspectral Frame Camera’, *The Photogrammetric Record*, vol. 31, no. 155, pp. 325–347, Sep. 2016, doi: 10.1111/phor.12153.

[109] E. Honkavaara, T. Rosnell, R. Oliveira, and A. Tommaselli, ‘Band registration of tuneable frame format hyperspectral UAV imagers in complex scenes’, *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 134, pp. 96–109, Dec. 2017, doi: 10.1016/j.isprsjprs.2017.10.014.

[110] K. Jacobsen, ‘Geometric Calibration of Space Remote Sensing Cameras for Efficient Processing’, p. 11.

[111] F. Remondino and C. Fraser, ‘Digital camera calibration methods: considerations and comparisons’, in *The Photogrammetric Record*, 2006, vol. 36 (5), pp. 266–272, Accessed: Nov. 14, 2019. [Online]. Available: http://doi.wiley.com/10.1111/j.1477-9730.2007.00418.x.

[112] R. Avtar and T. Watanabe, Eds., *Unmanned Aerial Vehicle: Applications in Agriculture and Environment*. Cham: Springer International Publishing, 2020.

[113] X. Yu, Y. Sun, A. Fang, W. Qi, and C. Liu, ‘Laboratory spectral calibration and radiometric calibration of hyper-spectral imaging spectrometer’, in *The 2014 2nd International Conference on Systems and Informatics (ICSAI 2014)*, Shanghai, China, Nov. 2014, pp. 871–875, doi: 10.1109/ICSAI.2014.7009407.

[114] Z. Gao, R. Jia, H. Zhang, Z. Xia, and W. Fang, ‘Simulation and Analysis of Spectral Response Function and Bandwidth of Spectrometer’, *International Journal of Aerospace Engineering*, vol. 2016, pp. 1–8, 2016, doi: 10.1155/2016/2759121.

[115] ‘Full width at half maximum’, *Wikipedia*. Nov. 03, 2020, Accessed: Nov. 06, 2020. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Full_width_at_half_maximum&oldid=986855772.

[116] M. Borengasser, W. S. Hungate, and R. Watkins, *Hyperspectral Remote Sensing: Principles and Applications*, 0 ed. CRC Press, 2007.

[117] Z. Malenovský, *Quantitative remote sensing of Norway spruce (Picea abies (L.) Karst.): Spectroscopy from needles to crowns to canopies*. Wageningen, 2006.

[118] H. Aasen, E. Honkavaara, A. Lucieer, and P. Zarco-Tejada, ‘Quantitative Remote Sensing at Ultra-High Resolution with UAV Spectroscopy: A Review of Sensor Technology, Measurement Procedures, and Data Correction Workflows’, *Remote Sensing*, vol. 10, no. 7, p. 1091, Jul. 2018, doi: 10.3390/rs10071091.

[119] S. Del Pozo, P. Rodríguez-Gonzálvez, D. Hernández-López, and B. Felipe-García, ‘Vicarious Radiometric Calibration of a Multispectral Camera on Board an Unmanned Aerial System’, *Remote Sensing*, vol. 6, no. 3, pp. 1918–1937, Feb. 2014, doi: 10.3390/rs6031918.

[120] J. Jablonski *et al.*, ‘Best practices in passive remote sensing VNIR hyperspectral system hardware calibrations’, Baltimore, Maryland, United States, May 2016, p. 986004, doi: 10.1117/12.2224022.

[121] K. J. Thome, ‘Ground-Look Radiometric Calibration Approaches for Remote Sensing Imagers in the Solar Reflective’, p. 6.

[122] M. Dinguirard and P. N. Slater, ‘Calibration of Space-Multispectral Imaging Sensors: A Review’, p. 12.

[123] E. Honkavaara and E. Khoramshahi, ‘Radiometric Correction of Close-Range Spectral Image Blocks Captured Using an Unmanned Aerial Vehicle with a Radiometric Block Adjustment’, *Remote Sensing*, vol. 10, no. 2, Art. no. 2, Feb. 2018, doi: 10.3390/rs10020256.

[124] T. Hakala *et al.*, ‘Direct Reflectance Measurements from Drones: Sensor Absolute Radiometric Calibration and System Tests for Forest Reflectance Characterization’, *Sensors*, vol. 18, no. 5, p. 1417, May 2018, doi: 10.3390/s18051417.

[125] S. Cao, B. Danielson, S. Clare, S. Koenig, C. Campos-Vargas, and A. Sanchez-Azofeifa, ‘Radiometric calibration assessments for UAS-borne multispectral cameras: Laboratory and field protocols’, *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 149, pp. 132–145, Mar. 2019, doi: 10.1016/j.isprsjprs.2019.01.016.

[126] P. N. Slater and S. F. Biggar, ‘Suggestions for Radiometric Calibration Coefficient Generation’, *Journal of Atmospheric and Oceanic Technology*, vol. 13, no. 2, pp. 376–382, Apr. 1996, doi: 10.1175/1520-0426(1996)013<0376:SFRCCG>2.0.CO;2.

[127] A. Burkart, S. Cogliati, A. Schickling, and U. Rascher, ‘A novel UAV-based ultra-light weight spectrometer for field spectroscopy’, *IEEE sensors journal*, vol. 14, no. 1, pp. 62–67, 2013.

[128] J. F. Burkhart *et al.*, ‘Unmanned aerial system nadir reflectance and MODIS nadir BRDF-adjusted surface reflectances intercompared over Greenland’, *The Cryosphere*, vol. 11, no. 4, pp. 1575–1589, Jul. 2017, doi: https://doi.org/10.5194/tc-11-1575-2017.

[129] L3HARRIS Geospatial, ‘ENVI - The Leading Geospatial Image Analysis Software’. https://www.harrisgeospatial.com/Software-Technology/ENVI (accessed Jun. 01, 2020).

[130] F. Iqbal, A. Lucieer, and K. Barry, ‘Simplified radiometric calibration for UAS-mounted multispectral sensor’, *European Journal of Remote Sensing*, vol. 51, no. 1, pp. 301–313, Jan. 2018, doi: 10.1080/22797254.2018.1432293.

[131] C. Wang and S. W. Myint, ‘A Simplified Empirical Line Method of Radiometric Calibration for Small Unmanned Aircraft Systems-Based Remote Sensing’, *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 8, no. 5, pp. 1876–1885, May 2015, doi: 10.1109/JSTARS.2015.2422716.

[132] H. Aasen, A. Burkart, A. Bolten, and G. Bareth, ‘Generating 3D hyperspectral information with lightweight UAV snapshot cameras for vegetation monitoring: From camera calibration to quality assurance’, *ISPRS*

Journal of Photogrammetry and Remote Sensing, vol. 108, pp. 245–259, Oct. 2015, doi: 10.1016/j.isprsjprs.2015.08.002.

[133] G. Yang *et al.*, ‘The DOM Generation and Precise Radiometric Calibration of a UAV-Mounted Miniature Snapshot Hyperspectral Imager’, *Remote Sensing*, vol. 9, no. 7, Art. no. 7, Jul. 2017, doi: 10.3390/rs9070642.

[134] H. Aasen and A. Bolten, ‘Multi-temporal high-resolution imaging spectroscopy with hyperspectral 2D imagers – From theory to application’, *Remote Sensing of Environment*, vol. 205, pp. 374–389, Feb. 2018, doi: 10.1016/j.rse.2017.10.043.

[135] C. L. Walthall, J. M. Norman, J. M. Welles, G. Campbell, and B. L. Blad, ‘Simple equation to approximate the bidirectional reflectance from vegetative canopies and bare soil surfaces’, *Appl. Opt., AO*, vol. 24, no. 3, pp. 383–387, Feb. 1985, doi: 10.1364/AO.24.000383.

[136] T. Nilson and A. Kuusk, ‘A reflectance model for the homogeneous plant canopy and its inversion’, *Remote Sensing of Environment*, vol. 27, no. 2, pp. 157–167, Feb. 1989, doi: 10.1016/0034-4257(89)90015-1.

[137] E. Honkavaara, T. Hakala, L. Markelin, T. Rosnell, H. Saari, and J. Mäkynen, ‘A Process for Radiometric Correction of UAV Image Blocks Verfahren zur radiometrischen Korrektur von UAV Bildblöcken’, *Photogrammetrie - Fernerkundung - Geoinformation*, vol. 2012, no. 2, pp. 115–127, May 2012, doi: 10.1127/1432-8364/2012/0106.

[138] J. D. Shepherd and J. R. Dymond, ‘Correcting satellite imagery for the variance of reflectance and illumination with topography’, *null*, vol. 24, no. 17, pp. 3503–3514, Jan. 2003, doi: 10.1080/01431160210154029.

[139] S. Jakob, R. Zimmermann, and R. Gloaguen, ‘The Need for Accurate Geometric and Radiometric Corrections of Drone-Borne Hyperspectral Data for Mineral Exploration: MEPHySTo—A Toolbox for Pre-Processing Drone-Borne Hyperspectral Data’, *Remote Sensing*, vol. 9, no. 1, p. 88, Jan. 2017, doi: 10.3390/rs9010088.

[140] K. R. M. Adeline, M. Chen, X. Briottet, S. K. Pang, and N. Paparoditis, ‘Shadow detection in very high spatial resolution aerial images: A comparative study’, *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 80, pp. 21–38, Jun. 2013, doi: 10.1016/j.isprsjprs.2013.02.003.

[141] K. D. Gremban, C. E. Thorpe, and T. Kanade, ‘Geometric camera calibration using systems of linear equations’, in *Proceedings. 1988 IEEE International Conference on Robotics and Automation*, Philadelphia, PA, USA, 1988, pp. 562–567, doi: 10.1109/ROBOT.1988.12111.

[142] D. C. Brown, ‘Close-Range Camera Calibration’, *Photogrammetric Eng.*, vol. 37, no. 8, pp. 855–866, 1971.

[143] J. Hieronymus, ‘Comparison of Methods for Geometric Camera Calibration’, p. 5, 2012.

[144] K. Kraus, I. A. Harley, and S. Kyle, *Photogrammetry, Advanced method and application*, 2nd ed., vol. 2. Bonn : Dümmler: Walter De Gruyter, 1997.

[145] P. Pazziani, ‘Progetto, realizzazione e calibrazione di un sistema di visione stereoscopica’, Università degli Studi di Roma “Tor Vergata”, Roma, 2011.

[146] M. Cramer, H.-J. Przybilla, and A. Zurhorst, ‘UAV CAMERAS: OVERVIEW AND GEOMETRIC CALIBRATION BENCHMARK’, *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*, vol. XLII-2/W6, pp. 85–92, Aug. 2017, doi: 10.5194/isprs-archives-XLII-2-W6-85-2017.

[147] I. Aicardi, ‘L’ utilizzo di smartphone per il posizionamento attraverso LiDAR 3D e fotogrammetria digitale’, *BollettinoSIFET*, no. 1, Art. no. 1, Apr. 2015.

[148] J.-Y. Bouguet, ‘Camera Calibration Toolbox for Matlab’, *Computational Vision at the California Institute of Technology*. http://www.vision.caltech.edu/bouguetj/calib_doc/ (accessed Nov. 14, 2019).

[149] ‘Agisoft Metashape’. https://www.agisoft.com/ (accessed Nov. 14, 2019).

[150] ‘Professional photogrammetry and drone mapping software | Pix4D’. https://www.pix4d.com/ (accessed Nov. 01, 2020).

[151] J. D. D. Junior, A. R. Backes, M. C. Escarpinati, B. C. S. Costa, and M. H. F. Avelar, ‘UAV-Multispectral Sensed Data Band Co-Registration Framework’, p. 6, 2020.

[152] E. Honkavaara *et al.*, ‘Processing and Assessment of Spectrometric, Stereoscopic Imagery Collected Using a Lightweight UAV Spectral Camera for Precision Agriculture’, *Remote Sensing*, vol. 5, no. 10, Art. no. 10, Oct. 2013, doi: 10.3390/rs5105006.

[153] R. Yasir, ‘Data driven multispectral image registration framework’, 2018.

[154] J. P. Jhan and J. Y. Rau, ‘A normalized SURF for multispectral image matching and band co-registration’, *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*, vol. XLII-2/W13, pp. 393–399, Jun. 2019, doi: 10.5194/isprs-archives-XLII-2-W13-393-2019.

[155] J.-P. Jhan, J.-Y. Rau, and N. Haala, ‘Robust and adaptive band-to-band image transform of UAS miniature multi-lens multispectral camera’, *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 137, pp. 47–60, Mar. 2018, doi: 10.1016/j.isprsjprs.2017.12.009.

[156] B. P. Banerjee, S. A. Raval, and P. J. Cullen, ‘Alignment of UAV-hyperspectral bands using keypoint descriptors in a spectrally complex environment’, *Remote Sensing Letters*, vol. 9, no. 6, pp. 524–533, Jun. 2018, doi: 10.1080/2150704X.2018.1446564.

[157] S. Leutenegger, M. Chli, and R. Y. Siegwart, ‘BRISK: Binary Robust invariant scalable keypoints’, in *2011 International Conference on Computer Vision*, Barcelona, Spain, Nov. 2011, pp. 2548–2555, doi: 10.1109/ICCV.2011.6126542.

[158] J. Matas, O. Chum, M. Urban, and T. Pajdla, ‘Robust wide-baseline stereo from maximally stable extremal regions’, *Image and Vision Computing*, vol. 22, no. 10, pp. 761–767, Sep. 2004, doi: 10.1016/j.imavis.2004.02.006.

[159] C. Harris and M. Stephens, ‘A Combined Corner and Edge Detector’, in *Procedings of the Alvey Vision Conference 1988*, Manchester, 1988, p. 23.1-23.6, doi: 10.5244/C.2.23.

[160] David G. Lowe, ‘Distinctive Image Features from Scale-Invariant Keypoints’, *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004, doi: 10.1023/B:VISI.0000029664.99615.94.

[161] H. Bay, T. Tuytelaars, and L. Van Gool, ‘SURF: Speeded Up Robust Features’, in *Computer Vision – ECCV 2006*, Berlin, Heidelberg, 2006, pp. 404–417, doi: 10.1007/11744023_32.

[162] P. F. Alcantarilla, A. Bartoli, and A. J. Davison, ‘KAZE Features’, in *Computer Vision – ECCV 2012*, vol. 7577, A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 214–227.

[163] E. Pastor, J. Lopez, and P. Royo, 'UAV Payload and Mission Control Hardware/Software Architecture', *IEEE Aerosp. Electron. Syst. Mag.*, vol. 22, no. 6, pp. 3–8, Jun. 2007, doi: 10.1109/MAES.2007.384074.

[164] L. Cwojdziński and M. Adamski, 'Power units and power supply systems in UAV', *Aviation*, vol. 18, no. 1, pp. 1–8, Apr. 2014, doi: 10.3846/16487788.2014.865938.

[165] A. Flodell and C. Christensson, 'Wildlife Surveillance Usinga UAV and Thermal Imagery', Master of Science, Linköping University, Sweden, 2016.

[166] S. Betti and C. Lodovisi, 'Studio e sviluppo di un protocollo di comunicazione per la raccolta dei dati sensoriali di bordo di un drone', p. 21.

[167] C. MAPIR, 'DJI Phantom 4/3 (PRO/ADV) Survey 3/2/1 Quad Camera Mount', *MAPIR CAMERA*, Dec. 04, 2020. https://www.mapir.camera/products/dji-phantom-4-and-3-mapir-survey-2-and-1-quad-camera-mount (accessed Dec. 04, 2020).

[168] DJI, 'Serie Matrice 200 V2. Scheda tecnica', *DJI Official*. https://www.dji.com/it/matrice-200-series-v2/info (accessed Dec. 04, 2020).

[169] DJI, 'Manifold 2', *DJI Official*, Dec. 04, 2020. https://www.dji.com/it/manifold-2 (accessed Dec. 04, 2020).

[170] Intel, 'Intel® Compute Stick STK2mv64CC. Specifiche dei prodotti', *Intel*, Dec. 04, 2020. https://ark.intel.com/content/www/it/it/ark/products/91979/intel-compute-stick-stk2mv64cc.html (accessed Dec. 04, 2020).

[171] D. Nicoletti, 'Progettazione di un POD per camera iperspettrale Senop su drone DJI matrice 210 v2', Bachelor's thesis, Politecnico di Torino, Torino, 2019.

[172] Senop, 'Senop - Optronics Hyperspectral'. https://senop.fi/optronics-hyperspectral (accessed Nov. 14, 2019).

[173] O. Pekkala, T. Pulli, A. Kokka, and E. Ikonen, 'Setup for characterising the spectral responsivity of Fabry–Pérot-interferometer-based hyperspectral cameras', *Metrologia*, vol. 56, no. 6, p. 065005, Dec. 2019, doi: 10.1088/1681-7575/ab3fd1.

[174] 'SOCET GXP Geospatial Intelligence Software', *Geospatial eXploitation Products*. https://www.geospatialexploitationproducts.com/content/socet-gxp/ (accessed Nov. 06, 2020).

[175] S. Walker, 'New Features in SOCET SET®', p. 6.

[176] L. Markelin *et al.*, 'Methodology for direct reflectance measurement from a drone: system description, radiometric calibration and latest results', *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*, vol. XLII–1, pp. 283–288, Sep. 2018, doi: 10.5194/isprs-archives-XLII-1-283-2018.

[177] A. M. G. Tommaselli, L. D. Santos, R. A. de Oliveira, A. Berveglieri, N. N. Imai, and E. Honkavaara, 'Refining the Interior Orientation of a Hyperspectral Frame Camera With Preliminary Bands Co-Registration', *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 12, no. 7, pp. 2097–2106, Jul. 2019, doi: 10.1109/JSTARS.2019.2911547.

[178] A. M. G. Tommaselli, A. Berveglieri, R. A. Oliveira, L. Y. Nagai, and E. Honkavaara, 'Orientation and calibration requirements for hyperpectral imaging using UAVs: a case study', *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XL-

3/W4, pp. 109–115, Mar. 2016, doi: 10.5194/isprsarchives-XL-3-W4-109-2016.

[179] M. Vakalopoulou and K. Karantzalos, ‘Automatic Descriptor-Based Co-Registration of Frame Hyperspectral Data’, *Remote Sensing*, vol. 6, no. 4, Art. no. 4, Apr. 2014, doi: 10.3390/rs6043409.

[180] A. M. G. Tommaselli, R. A. Oliveira, L. Y. Nagai, N. N. Imai, E. Honkavaara, and T. Hakala, ‘Assessment of bands coregistration of a light-weight spectral frame camera for UAV’, p. 4.

[181] H. Saari *et al.*, ‘Novel miniaturized hyperspectral sensor for UAV and space applications’, Berlin, Germany, Sep. 2009, p. 74741M, doi: 10.1117/12.830284.

[182] Senop, ‘Rikola Hyperspectral camera Datasheet’, *Senop*, 2018. https://senop.fi/files/rikola/pdf/Hyperspectral+Camera_Datasheet.pdf (accessed Jun. 28, 2020).

[183] A. M. G. Tommaselli, L. D. Santos, R. A. de Oliveira, and E. Honkavaara, ‘Refining the Geometric Calibration of a Hiperspectral Frame Camera with Preliminary Bands Coregistration’, in *IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium*, Valencia, Spain, Jul. 2018, pp. 6468–6471, doi: 10.1109/IGARSS.2018.8519058.

[184] A. Rissanen *et al.*, ‘VTT’s Fabry-Perot interferometer technologies for hyperspectral imaging and mobile sensing applications’, San Francisco, California, United States, Feb. 2017, p. 101160I, doi: 10.1117/12.2255950.

[185] A. Vaccaro, ‘Il telerilevamento iperspettrale: messa a punto e calibrazione di una camera iperspettrale’, *Unpublished*, 2017, doi: 10.13140/rg.2.2.32558.41288.

[186] Senop, ‘Rikola Hyperspectral Imager. Manual’. 2017.

[187] L3 Harris Geospatial, ‘Docs Center - Using ENVI- Atmospheric Correction’, *L3 Harris Geospatial Solution*. https://www.harrisgeospatial.com/docs/AtmosphericCorrection.html#empirical_line_calibration (accessed Jul. 04, 2020).

[188] M. CAMERA, ‘MAPIR Camera Reflectance Calibration Ground Target Package (V2)’, *MAPIR CAMERA*. https://www.mapir.camera/products/mapir-camera-reflectance-calibration-ground-target-package-v2 (accessed Nov. 06, 2020).

[189] ‘NDVI | field | reflectance calibration targets | plates for agriculture’. http://www.mosaicmill.com/products_other/reflectance_targets.html (accessed Nov. 06, 2020).

[190] M. A. Musci, I. Aicardi, P. Dabove, and A. M. Lingua, ‘Reliability of the geometric calibration of a hyperspectral frame camera’, *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*, vol. XLII-2/W13, pp. 1701–1707, Jun. 2019, doi: 10.5194/isprs-archives-XLII-2-W13-1701-2019.

[191] K. Fekete and P. Schrott, ‘Qualification of close-range photogrammetry cameras by average image coordinates RMS Error vs. object distance function’, p. 4, 2008.

[192] MATLAB, ‘Documentation. Configurations for intensity-based registration - MATLAB imregconfig’, *MathWorks Italia*, Dec. 07, 2020. https://it.mathworks.com/help/images/ref/imregconfig.html (accessed Dec. 07, 2020).

[193] T. M. Lillesand, R. W. Kiefer, and J. Chipman, *Remote Sensing and Image Interpretation*, 7th Edition. Wiley, 2015.

[194] J. Gao, *Digital analysis of remotely sensed imagery*. New York: McGraw Hill, 2009.

[195] S. Behnke and R. Rojas, 'Neural abstraction pyramid: a hierarchical image understanding architecture', in *1998 IEEE International Joint Conference on Neural Networks Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98CH36227)*, Anchorage, AK, USA, 1998, vol. 2, pp. 820–825, doi: 10.1109/IJCNN.1998.685873.

[196] S. Behnke, *Hierarchical Neural Networks for Image Interpretation*, vol. 2766. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003.

[197] S. E. Umbaugh, *Computer imaging: digital image analysis and processing*. CRC press, 2005.

[198] S. G. Small and L. Medsker, 'Review of information extraction technologies and applications', *Neural Comput & Applic*, vol. 25, no. 3–4, pp. 533–548, Sep. 2014, doi: 10.1007/s00521-013-1516-6.

[199] K. V. Kale, M. M. Solankar, D. B. Nalawade, R. K. Dhumal, and H. R. Gite, 'A Research Review on Hyperspectral Data Processing and Analysis Algorithms', *Proc. Natl. Acad. Sci., India, Sect. A Phys. Sci.*, vol. 87, no. 4, pp. 541–555, Dec. 2017, doi: 10.1007/s40010-017-0433-y.

[200] H. Petersson, D. Gustafsson, and D. Bergstrom, 'Hyperspectral image analysis using deep learning — A review', in *2016 Sixth International Conference on Image Processing Theory, Tools and Applications (IPTA)*, Oulu, Finland, Dec. 2016, pp. 1–6, doi: 10.1109/IPTA.2016.7820963.

[201] R. E. Bellman, *Adaptive Control Processes. A Guided Tour*. NJ, USA, 1961.

[202] G. Hughes, 'On the mean accuracy of statistical pattern recognizers', *IEEE Transactions on Information Theory*, vol. 14, no. 1, pp. 55–63, Jan. 1968, doi: 10.1109/TIT.1968.1054102.

[203] D. A. Landgrebe, *Signal Theory Methods in Multispectral Remote Sensing*. John Wiley & Sons, 2005.

[204] A. Salimi, M. Ziaii, A. Amiri, M. Hosseinjani Zadeh, S. Karimpouli, and M. Moradkhani, 'Using a Feature Subset Selection method and Support Vector Machine to address curse of dimensionality and redundancy in Hyperion hyperspectral data classification', *The Egyptian Journal of Remote Sensing and Space Science*, vol. 21, no. 1, pp. 27–36, Apr. 2018, doi: 10.1016/j.ejrs.2017.02.003.

[205] L. Yu and H. Liu, 'Efficient Feature Selection via Analysis of Relevance and Redundancy', vol. 5, p. 20, 2004.

[206] L. Ladha and T. Deepa, 'Feature selection methods and algorithms', vol. 3, no. 5, p. 11, 2011.

[207] R. Zebari, A. Abdulazeez, D. Zeebaree, D. Zebari, and J. Saeed, 'A Comprehensive Review of Dimensionality Reduction Techniques for Feature Selection and Feature Extraction', *JASTT*, vol. 1, no. 2, pp. 56–70, May 2020, doi: 10.38094/jastt1224.

[208] Z. M. Hira and D. F. Gillies, 'A Review of Feature Selection and Feature Extraction Methods Applied on Microarray Data', *Advances in Bioinformatics*, vol. 2015, pp. 1–13, 2015, doi: 10.1155/2015/198363.

[209] X. Huang, L. Wu, and Y. Ye, ‘A Review on Dimensionality Reduction Techniques’, *Int. J. Patt. Recogn. Artif. Intell.*, vol. 33, no. 10, p. 1950017, Sep. 2019, doi: 10.1142/S0218001419500174.

[210] J. Han, M. Kamber, and J. Pei, *Data Mining. Concepts and Techniques*, 3rd Edition. Morgan Kaufmann Publishers, 2012.

[211] Y. Zhou, R. Zhang, S. Wang, and F. Wang, ‘Feature Selection Method Based on High-Resolution Remote Sensing Images and the Effect of Sensitive Features on Classification Accuracy’, *Sensors (Basel)*, vol. 18, no. 7, Jun. 2018, doi: 10.3390/s18072013.

[212] V. Bolón-Canedo and B. Remeseiro, ‘Feature selection in image analysis: a survey’, *Artif Intell Rev*, vol. 53, no. 4, pp. 2905–2931, Apr. 2020, doi: 10.1007/s10462-019-09750-3.

[213] S. Khalid, T. Khalil, and S. Nasreen, ‘A survey of feature selection and feature extraction techniques in machine learning’, in *2014 Science and Information Conference*, London, UK, Aug. 2014, pp. 372–378, doi: 10.1109/SAI.2014.6918213.

[214] B. Mojaradi, H. Abrishami-Moghaddam, M. J. V. Zoej, and R. P. W. Duin, ‘Dimensionality Reduction of Hyperspectral Data via Spectral Feature Extraction’, *IEEE Trans. Geosci. Remote Sensing*, vol. 47, no. 7, pp. 2091–2105, Jul. 2009, doi: 10.1109/TGRS.2008.2010346.

[215] K. Pearson, ‘On lines and planes of closest fit to systems of points in space’, *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 2, no. 11, pp. 559–572, Nov. 1901, doi: 10.1080/14786440109462720.

[216] B. Schölkopf, A. Smola, and K.-R. Müller, ‘Nonlinear Component Analysis as a Kernel Eigenvalue Problem’, *Neural Computation*, vol. 10, no. 5, pp. 1299–1319, Jul. 1998, doi: 10.1162/089976698300017467.

[217] S. Wold, M. Sjöström, and L. Eriksson, ‘PLS-regression: a basic tool of chemometrics’, *Chemometrics and Intelligent Laboratory Systems*, vol. 58, no. 2, pp. 109–130, Oct. 2001, doi: 10.1016/S0169-7439(01)00155-1.

[218] P. Comon, ‘Independent component analysis, A new concept?’, *Signal Processing*, vol. 36, no. 3, pp. 287–314, Apr. 1994, doi: 10.1016/0165-1684(94)90029-9.

[219] T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning: Data mining, inference, and prediction*, 2nd ed. New York: Springer, 2009.

[220] X. Zhang, F. Liu, Y. He, and X. Li, ‘Application of hyperspectral imaging and chemometric calibrations for variety discrimination of maize seeds’, *Sensors (Basel)*, vol. 12, no. 12, pp. 17234–17246, Dec. 2012, doi: 10.3390/s121217234.

[221] W. Zhang, X. Li, and L. Zhao, ‘Band Priority Index: A Feature Selection Framework for Hyperspectral Imagery’, *Remote Sensing*, vol. 10, no. 7, p. 1095, Jul. 2018, doi: 10.3390/rs10071095.

[222] G. Camps-Valls, D. Tuia, L. Gómez-Chova, S. Jiménez, and J. Malo, ‘Remote Sensing Image Processing’, *Synthesis Lectures on Image, Video, and Multimedia Processing*, vol. 5, no. 1, pp. 1–192, Dec. 2011, doi: 10.2200/S00392ED1V01Y201107IVM012.

[223] P. Preet, S. S. Batra, and Jayadeva, ‘Feature Selection for classification of hyperspectral data by minimizing a tight bound on the VC dimension’,

arXiv:1509.08112 [cs], Sep. 2015, Accessed: Jun. 15, 2020. [Online]. Available: http://arxiv.org/abs/1509.08112.

[224] 'Feature Selection Methods | Machine Learning', *Analytics Vidhya*, Dec. 01, 2016. https://www.analyticsvidhya.com/blog/2016/12/introduction-to-feature-selection-methods-with-an-example-or-how-to-select-the-right-variables/ (accessed Oct. 08, 2020).

[225] A. Jovic, K. Brkic, and N. Bogunovic, 'A review of feature selection methods with applications', in *2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, Opatija, Croatia, May 2015, pp. 1200–1205, doi: 10.1109/MIPRO.2015.7160458.

[226] M. Walowe Mwadulo, 'A Review on Feature Selection Methods For Classification Tasks', *IJCATR*, vol. 5, no. 6, pp. 395–402, Jun. 2016, doi: 10.7753/IJCATR0506.1013.

[227] W. Sun and Q. Du, 'Hyperspectral Band Selection: A Review', *IEEE Geosci. Remote Sens. Mag.*, vol. 7, no. 2, pp. 118–139, Jun. 2019, doi: 10.1109/MGRS.2019.2911100.

[228] Chein-I Chang, Qian Du, Tzu-Lung Sun, and M. L. G. Althouse, 'A joint band prioritization and band-decorrelation approach to band selection for hyperspectral image classification', *IEEE Trans. Geosci. Remote Sensing*, vol. 37, no. 6, pp. 2631–2641, Nov. 1999, doi: 10.1109/36.803411.

[229] F. Song, Z. Guo, and D. Mei, 'Feature Selection Using Principal Component Analysis', in *2010 International Conference on System Science, Engineering Design and Manufacturing Informatization*, Yichang, China, Nov. 2010, pp. 27–30, doi: 10.1109/ICSEM.2010.14.

[230] M. A. Kramer, 'Nonlinear principal component analysis using autoassociative neural networks', *AIChE journal*, vol. 37, no. 2, pp. 233–243, 1991.

[231] J. Estornell, J. M. Martı-Gavila, M. T. Sebastia, and J. Mengual, 'Principal component analysis applied to remote sensing', p. 8.

[232] M. Fauvel, J. Chanussot, and J. A. Benediktsson, 'Kernel Principal Component Analysis for the Classification of Hyperspectral Remote Sensing Data over Urban Areas', *EURASIP J. Adv. Signal Process.*, vol. 2009, no. 1, p. 783194, Dec. 2009, doi: 10.1155/2009/783194.

[233] N. Torbick and B. Becker, 'Evaluating Principal Components Analysis for Identifying Optimal Bands Using Wetland Hyperspectral Measurements From the Great Lakes, USA', *Remote Sensing*, vol. 1, no. 3, pp. 408–417, Aug. 2009, doi: 10.3390/rs1030408.

[234] I. T. Jolliffe, *Principal Component Analysis*, 2ed. Springer, 2002.

[235] R. B. Cattell, 'The Scree Test For The Number Of Factors', *Multivariate Behavioral Research*, vol. 1, no. 2, pp. 245–276, Apr. 1966, doi: 10.1207/s15327906mbr0102_10.

[236] F. Csillag, L. Pásztor, and L. L. Biehl, 'Spectral band selection for the characterization of salinity status of soils', *Remote sensing of environment*, vol. 43, no. 3, pp. 231–242, 1993.

[237] M. Kim *et al.*, 'Deep Learning in Medical Imaging', *Neurospine*, vol. 16, no. 4, pp. 657–668, Dec. 2019, doi: 10.14245/ns.1938396.198.

[238] Hae Jong Seo and P. Milanfar, 'Training-Free, Generic Object Detection Using Locally Adaptive Regression Kernels', *IEEE Trans. Pattern Anal. Mach.*

Intell., vol. 32, no. 9, pp. 1688–1704, Sep. 2010, doi: 10.1109/TPAMI.2009.153.

[239] C. A. Glasbey and G. W. Horgan, 'Image Analysis for the biological sciences', p. 26.

[240] T. T. Santos, L. L. de Souza, A. A. dos Santos, and S. Avila, 'Grape detection, segmentation and tracking using deep neural networks and three-dimensional association', *Computers and Electronics in Agriculture*, vol. 170, p. 105247, Mar. 2020, doi: 10.1016/J.COMPAG.2020.105247.

[241] J. A. Richards and X. Jia, *Remote sensing digital image analysis: an introduction*, 4th ed. Berlin: Springer, 2006.

[242] R. Divyanshu and T. Deepti, 'A Review on Image Classification Approaches and Techniques', *IJSRSET - International Journal of Scientific Research in Science, Engineering and Technology*, vol. 2, no. 6, 2016.

[243] S. W. Myint, V. Mesev, D. Quattrochi, and E. A. Wentz, '10. Urban Image Classification: Per-pixel Classifiers, Sub-pixel Analysis, Object-based Image Analysis, and Geospatial Methods', p. 42.

[244] Q. Weng and X. Hu, 'Medium Spatial Resolution Satellite Imagery for Estimating and Mapping Urban Impervious Surfaces Using LSMA and ANN', *IEEE Transactions on Geoscience and Remote Sensing*, vol. 46, no. 8, pp. 2397–2406, Aug. 2008, doi: 10.1109/TGRS.2008.917601.

[245] S. W. Myint and G. S. Okin, 'Modelling land-cover types using multiple endmember spectral mixture analysis in a desert city', p. 22.

[246] 'Thomas Blaschke, Stefan Lang, Geoffrey J. Hay - Object-Based Image Analysis_ Spatial Concepts for Knowledge-Driven Remote Sensing Applications (Lecture Notes in Geoinformation and Cartography) (2008, Springer) - libgen.lc.pdf'. .

[247] A. Tzotsos and D. Argialas, 'Support Vector Machine Classification for Object-Based Image Analysis', in *Object-Based Image Analysis: Spatial Concepts for Knowledge-Driven Remote Sensing Applications*, T. Blaschke, S. Lang, and G. J. Hay, Eds. Berlin, Heidelberg: Springer, 2008, pp. 663–677.

[248] D. C. Duro, S. E. Franklin, and M. G. Dubé, 'A comparison of pixel-based and object-based image analysis with selected machine learning algorithms for the classification of agricultural landscapes using SPOT-5 HRG imagery', *Remote Sensing of Environment*, vol. 118, pp. 259–272, Mar. 2012, doi: 10.1016/j.rse.2011.11.020.

[249] N. S. Altman, 'An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression', *The American Statistician*, vol. 46, no. 3, pp. 175–185, Aug. 1992, doi: 10.1080/00031305.1992.10475879.

[250] G. E. Hinton, S. Osindero, and Y.-W. Teh, 'A Fast Learning Algorithm for Deep Belief Nets', *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, Jul. 2006, doi: 10.1162/neco.2006.18.7.1527.

[251] A. Nandy and M. Biswas, *Reinforcement Learning*. Berkeley, CA: Apress, 2018.

[252] A. V. Bernstein and E. V. Burnaev, 'Reinforcement learning in computer vision', in *Tenth International Conference on Machine Vision (ICMV 2017)*, Vienna, Austria, Apr. 2018, p. 58, doi: 10.1117/12.2309945.

[253] R. Furuta, N. Inoue, and T. Yamasaki, 'PixelRL: Fully Convolutional Network with Reinforcement Learning for Image Processing',

arXiv:1912.07190 [cs, eess], Dec. 2019, Accessed: Jun. 14, 2020. [Online]. Available: http://arxiv.org/abs/1912.07190.

[254] D. Li, H. Wu, J. Zhang, and K. Huang, ‘A2-RL: Aesthetics Aware Reinforcement Learning for Image Cropping’, *arXiv:1709.04595 [cs]*, Mar. 2018, Accessed: Jun. 14, 2020. [Online]. Available: http://arxiv.org/abs/1709.04595.

[255] Z. Jie, X. Liang, J. Feng, X. Jin, W. Lu, and S. Yan, ‘Tree-Structured Reinforcement Learning for Sequential Object Localization’, in *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, Eds. Curran Associates, Inc., 2016, pp. 127–135.

[256] Ph. D. Thenkabail, Ed., *Remotely Sensed Data Characterization, Classification, and Accuracies*, 0 ed. CRC Press, 2015.

[257] J. Dong, Q. Chen, S. Yan, and A. Yuille, ‘Towards Unified Object Detection and Semantic Segmentation’, in *Computer Vision – ECCV 2014*, vol. 8693, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham: Springer International Publishing, 2014, pp. 299–314.

[258] P. Thanh Noi and M. Kappas, ‘Comparison of Random Forest, k-Nearest Neighbor, and Support Vector Machine Classifiers for Land Cover Classification Using Sentinel-2 Imagery’, *Sensors*, vol. 18, no. 2, p. 18, Dec. 2017, doi: 10.3390/s18010018.

[259] M. O. Ngadi and L. Liu, ‘Hyperspectral Image Processing Techniques’, in *Hyperspectral Imaging for Food Quality Analysis and Control*, Elsevier, 2010, pp. 99–127.

[260] T. Liu, A. Abd-Elrahman, J. Morton, and V. L. Wilhelm, ‘Comparing fully convolutional networks, random forest, support vector machine, and patch-based deep convolutional neural networks for object-based wetland mapping using images from small unmanned aircraft system’, *GIScience & remote sensing*, vol. 55, no. 2, pp. 243–264, 2018.

[261] J. A. Gualtieri and R. F. Cromp, ‘Support vector machines for hyperspectral remote sensing classification’, in *27th AIPR Workshop: Advances in Computer-Assisted Recognition*, Jan. 1999, vol. 3584, pp. 221–232, doi: 10.1117/12.339824.

[262] F. Melgani and L. Bruzzone, ‘Classification of hyperspectral remote sensing images with support vector machines’, *IEEE Trans. Geosci. Remote Sensing*, vol. 42, no. 8, pp. 1778–1790, Aug. 2004, doi: 10.1109/TGRS.2004.831865.

[263] Q. Feng, J. Liu, and J. Gong, ‘UAV Remote Sensing for Urban Vegetation Mapping Using Random Forest and Texture Analysis’, *Remote Sensing*, vol. 7, no. 1, Art. no. 1, Jan. 2015, doi: 10.3390/rs70101074.

[264] C. Zhang and Y. Ma, Eds., *Ensemble Machine Learning*. Boston, MA: Springer US, 2012.

[265] K. Li, G. Wan, G. Cheng, L. Meng, and J. Han, ‘Object detection in optical remote sensing images: A survey and a new benchmark’, *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 159, pp. 296–307, Jan. 2020, doi: 10.1016/j.isprsjprs.2019.11.023.

[266] R. Girshick, ‘Fast R-CNN’, *arXiv:1504.08083 [cs]*, Sep. 2015, Accessed: Apr. 17, 2020. [Online]. Available: http://arxiv.org/abs/1504.08083.

[267] J. Huang *et al.*, 'Speed/accuracy trade-offs for modern convolutional object detectors', *arXiv:1611.10012 [cs]*, Apr. 2017, Accessed: May 01, 2020. [Online]. Available: http://arxiv.org/abs/1611.10012.

[268] L. Liu *et al.*, 'Deep Learning for Generic Object Detection: A Survey', *arXiv:1809.02165 [cs]*, Aug. 2019, Accessed: Apr. 17, 2020. [Online]. Available: http://arxiv.org/abs/1809.02165.

[269] M. A. Musci, C. Persello, and A. M. Lingua, 'UAV images and Deep-Learning algorithms for detecting Flavescence Dorée disease in grapevine orchards', in *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Aug. 2020, vol. XLIII-B3-2020, pp. 1483–1489, doi: https://doi.org/10.5194/isprs-archives-XLIII-B3-2020-1483-2020.

[270] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, 'The Pascal Visual Object Classes (VOC) Challenge', *Int J Comput Vis*, vol. 88, no. 2, pp. 303–338, Jun. 2010, doi: 10.1007/s11263-009-0275-4.

[271] N. Craswell and S. Robertson, 'Average Precision at n', in *Encyclopedia of Database Systems*, L. LIU and M. T. ÖZSU, Eds. Boston, MA: Springer US, 2009, pp. 193–194.

[272] SEI Project Consortium, 'SEI Project', *SEI Project*. https://www.sei-project.eu/ (accessed Jul. 07, 2020).

[273] A. Falcone *et al.*, 'Spectral Evidence of Ice on Aircrafts – The SEI Project', p. 5.

[274] European Organisation for the Safety of Air Navigation and The Flight Safety Foundation, 'Aircraft Ground De/Anti-Icing', *SKYbrary Aviation Safety*. https://www.skybrary.aero/index.php/Aircraft_Ground_De/Anti-Icing (accessed Jun. 28, 2020).

[275] International Civil Aviation Organization (ICAO), 'Aircraft Ground De-Icing/Anti-Icing Operations', *ICAO United Aviation, A united nations specialized agency*. https://www.icao.int/safety/airnavigation/OPS/Pages/Aircraft-Ground-De-IcingAnti-Icing-Operations.aspx (accessed Jun. 25, 2020).

[276] S. Castro, L. C. Davis, and L. E. Erickson, 'Plant-Enhanced Remediation of Glycol-Based Aircraft Deicing Fluids', *Practice Periodical of Hazardous, Toxic, and Radioactive Waste Management*, vol. 5, no. 3, pp. 141–152, Jul. 2001, doi: 10.1061/(ASCE)1090-025X(2001)5:3(141).

[277] P. Rizk, N. Al Saleh, R. Younes, A. Ilinca, and J. Khoder, 'Hyperspectral imaging applied for the detection of wind turbine blade damage and icing', *Remote Sensing Applications: Society and Environment*, vol. 18, p. 100291, Apr. 2020, doi: 10.1016/j.rsase.2020.100291.

[278] C. E. Bassey and G. Simpson, 'Aircraft Ice Detection using Time Domain Reflectometry with Coplanar Sensors', *2007 IEEE Aerospace Conference*, 2007, doi: 10.1109/AERO.2007.352857.

[279] A. Troiano, E. Pasero, and L. Mesin, 'An innovative water and ice detection system for monitoring road and runway surfaces', in *6th Conference on Ph.D. Research in Microelectronics & Electronics*, Technische Universität, Berlin, Jul. 2010, p. 5, [Online]. Available: https://ieeexplore.ieee.org/document/5587181.

[280] J.-M. Breda, J. Choisnet, and M. Naim, 'Optical device for the detection of icing conditions on aircraft', US6091335A, Jul. 18, 2000.

[281] X. Gong and S. Bansmer, ‘3-D ice shape measurements using mid-infrared laser scanning’, *Opt. Express*, vol. 23, no. 4, p. 4908, Feb. 2015, doi: 10.1364/OE.23.004908.

[282] S. Nakauchi, K. Nishino, and T. Yamashita, ‘Selection of optimal combinations of band-pass filters for ice detection by hyperspectral imaging’, *Opt. Express*, vol. 20, no. 2, p. 986, Jan. 2012, doi: 10.1364/OE.20.000986.

[283] N. Huber, S. R. Kalidindi, B. Klusemann, and C. J. Cyron, *Machine Learning and Data Mining in Materials Science*. Frontiers Media SA, 2020.

[284] G. Camps-Valls and L. Bruzzone, ‘Kernel-based methods for hyperspectral image classification’, *IEEE Transactions on Geoscience and Remote Sensing*, vol. 43, no. 6, pp. 1351–1362, Jun. 2005, doi: 10.1109/TGRS.2005.846154.

[285] E. M. Adam, O. Mutanga, D. Rugege, and R. Ismail, ‘Discriminating the papyrus vegetation (Cyperus papyrus L.) and its co-existent species using random forest and hyperspectral data resampled to HYMAP’, *International Journal of Remote Sensing*, vol. 33, no. 2, pp. 552–569, Jan. 2012, doi: 10.1080/01431161.2010.543182.

[286] Federal Aviation Administration, United States Department of Transportation, ‘Standardized International Aircraft Ground Deice Program (SIAGDP)’, *Federal Aviation Administration, United States Department of Transportation*. https://www.faa.gov/other_visit/aviation_industry/airline_operators/airline_safety/deicing/media/standardized_international_ground_deice_program.pdf (accessed Jun. 25, 2020).

[287] G. A. Hoover, ‘Aircraft Ice Detectors and Related Technologies for Onground and Inflight Applications’, U.S. Department of Transportation Fedaral Aviation Administration Technical Center Atlantic City International Airport, NJ 08405, Galaxy Scientific Corporation 2500 English Creek Avenue, Building 11 Pleasantville, NJ 08232, Technical Report DOT/FAA/CT-92/27, Apr. 1993. Accessed: Jun. 26, 2020. [Online]. Available: https://apps.dtic.mil/dtic/tr/fulltext/u2/a266273.pdf.

[288] G. Mingione, M. Barocco, E. Denti, and F. G. Bindi, ‘Flight in icing conditions’, Direction gnrale de l’aviation civile (DGAC), 1997.

[289] Federal Aviation Administration, United States Department of Transportation, ‘AC 00-6A - Aviation Weather For Pilots and Flight Operations Personnel’. asa PUBLICATIONS, 1975, Accessed: Jul. 30, 2020. [Online]. Available: https://www.faa.gov/documentLibrary/media/Advisory_Circular/AC%2000-6A%20Chap%2010-12.pdf.

[290] Federal Aviation Administration, United States Department of Transportation, ‘AC 00-6B - Aviation Weather’. Aug. 23, 2016, Accessed: Jul. 29, 2020. [Online]. Available: https://www.faa.gov/documentlibrary/media/advisory_circular/ac_00-6b.pdf.

[291] ‘Icing Hazards’. https://www.weather.gov/source/zhu/ZHU_Training_Page/icing_stuff/icing/icing.htm (accessed Jun. 28, 2020).

[292] Universal Weather and Aviation, Inc., ‘Aircraft Icing and How It Affects Your Flight’, *Universal® Operational Insight Blog*, Jul. 30, 2013. https://www.universalweather.com/blog/aircraft-icing-and-how-it-affects-your-flight/ (accessed Dec. 22, 2020).

[293] American Meteorological Society, ‘Rime’, *Glossary of Meteorology*, Apr. 25, 2012. http://glossary.ametsoc.org/wiki/Rime (accessed Jul. 29, 2020).

[294] American Meteorological Society, ‘Glaze’, *Glossary of Meteorology*, Feb. 20, 2012. http://glossary.ametsoc.org/wiki/Glaze (accessed Jul. 29, 2020).

[295] Geomatics Group of Politecnico di Torino, ‘Our Labs- Geomatics Labs’. https://areeweb.polito.it/geomatics_lab/en/geomatic-lab/ (accessed Jul. 04, 2020).

[296] ESRI, ‘Composite Bands—Data Management toolbox’, *ArcGIS Pro*. https://pro.arcgis.com/en/pro-app/tool-reference/data-management/composite-bands.htm (accessed Jul. 05, 2020).

[297] ESRI, ‘Principal Components’, *ArcGIS Pro*. https://pro.arcgis.com/en/pro-app/tool-reference/spatial-analyst/principal-components.htm (accessed Jul. 03, 2020).

[298] M. A. Musci, L. Mazzara, and A. M. Lingua, ‘Ice Detection on Aircraft Surface Using Machine Learning Approaches Based on Hyperspectral and Multispectral Images’, *Drones*, vol. 4, no. 3, Art. no. 3, Sep. 2020, doi: 10.3390/drones4030045.

[299] ESRI, ‘An overview of the Segmentation and Classification toolset’, *ArcGIS Pro*. https://pro.arcgis.com/en/pro-app/tool-reference/spatial-analyst/an-overview-of-the-segmentation-and-classification-tools.htm (accessed Jun. 30, 2020).

[300] ESRI, ‘Classify Raster’, *ArcGIS Pro*. https://pro.arcgis.com/en/pro-app/tool-reference/spatial-analyst/classify-raster.htm (accessed Jul. 05, 2020).

[301] I. Aicardi, A. Lingua, L. Mazzara, M. A. Musci, and G. Rizzo, ‘Ice detection on airplane wings using a photogrammetric point cloud: a simulation’, *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XLIII-B2-2020, pp. 183–189, Aug. 2020, doi: 10.5194/isprs-archives-XLIII-B2-2020-183-2020.

[302] J. Smithers, ‘Sustainable Agriculture: Interpretations, Analyses and Prospects’, *SUSTAINABLE AGRICULTURE*, no. Canadian Journal of Regional Science, p. 14, 1993.

[303] E. U. Eurostat, ‘Agriculture, forestry and fishery statistics’, *Eurostat, Statistics Explained*, Oct. 20, 2020. https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Performance_of_the_agricultural_sector#Resource_performance_of_the_agricultural_sector (accessed Jan. 06, 2021).

[304] (JRC) Joint Research Centre of the European Commission, (MARS) Monitoring Agriculture Resources Unit H04, P. J. Zarco-Tejada, N. Hubbard, and P. Loudjani, ‘Precision agriculture: an opportunity for EU farmers: potential support with the CAP 2014-2020’, *AGRICULTURE AND RURAL DEVELOPMENT*, p. 56, Jun. 2014.

[305] J. Zoto, M. A. Musci, A. Khaliq, M. Chiaberge, and I. Aicardi, ‘Automatic Path Planning for Unmanned Ground Vehicle Using UAV Imagery’, 2020, pp. 223–230.

[306] A. Khaliq, M. A. Musci, and M. Chiaberge, ‘Analyzing relationship between maize height and spectral indices derived from remotely sensed multispectral imagery’, in *2018 IEEE Applied Imagery Pattern Recognition Workshop (AIPR)*, Oct. 2018, pp. 1–5, doi: 10.1109/AIPR.2018.8707373.

[307] A. Khaliq, M. A. Musci, and M. Chiaberge, *Understanding effects of atmospheric variables on spectral vegetation indices derived from satellite based time series of multispectral images*. 2018, p. 4.

[308] S. Khanal, K. Kc, J. P. Fulton, S. Shearer, and E. Ozkan, 'Remote Sensing in Agriculture—Accomplishments, Limitations, and Opportunities', *Remote Sensing*, vol. 12, no. 22, Art. no. 22, Jan. 2020, doi: 10.3390/rs12223783.

[309] W. Zhao, W. Yamada, T. Li, M. Digman, and T. Runge, 'Augmenting Crop Detection for Precision Agriculture with Deep Visual Transfer Learning—A Case Study of Bale Detection', *Remote Sensing*, vol. 13, no. 1, Art. no. 1, Jan. 2021, doi: 10.3390/rs13010023.

[310] E.-C. Oerke, 'Remote Sensing of Diseases', *Annu. Rev. Phytopathol.*, vol. 58, no. 1, pp. 225–252, Aug. 2020, doi: 10.1146/annurev-phyto-010820-012832.

[311] A. Matese and S. F. D. Gennaro, 'Technology in precision viticulture: a state of the art review', *International Journal of Wine Research*, May 18, 2015. https://www.dovepress.com/technology-in-precision-viticulture-a-state-of-the-art-review-peer-reviewed-article-IJWR (accessed May 03, 2020).

[312] F. Mazzetto, A. Calcante, A. Mena, and A. Vercesi, 'Integration of optical and analogue sensors for monitoring canopy health and vigour in precision viticulture', *Precision Agric*, vol. 11, no. 6, pp. 636–649, Dec. 2010, doi: 10.1007/s11119-010-9186-1.

[313] C. Micheloni, 'EIP-AGRI Focus Group . Diseases and pests in viticulture', *EIP-AGRI Focus Group . Diseases and pests in viticulture*, no. eip-agri, p. 18, 2017.

[314] J. Chuche and D. Thiéry, 'Biology and ecology of the Flavescence dorée vector Scaphoideus titanus: a review', *Agron. Sustain. Dev.*, vol. 34, no. 2, pp. 381–403, Apr. 2014, doi: 10.1007/s13593-014-0208-7.

[315] J. Zhou, L. R. Khot, H. Y. Bahlol, R. Boydston, and P. N. Miklas, 'Evaluation of ground, proximal and aerial remote sensing technologies for crop stress monitoring', *IFAC-PapersOnLine*, vol. 49, no. 16, pp. 22–26, Jan. 2016, doi: 10.1016/j.ifacol.2016.10.005.

[316] J. Hruška *et al.*, 'Evaluation of machine learning techniques in vine leaves disease detection: a preliminary case study on Flavescence Dorée', *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*, vol. XLII-3/W8, pp. 151–156, Aug. 2019, doi: 10.5194/isprs-archives-XLII-3-W8-151-2019.

[317] J. Albetis *et al.*, 'Detection of Flavescence dorée Grapevine Disease Using Unmanned Aerial Vehicle (UAV) Multispectral Imagery', *Remote Sensing*, vol. 9, no. 4, p. 308, Mar. 2017, doi: 10.3390/rs9040308.

[318] E. M. Abdel-Rahman, O. Mutanga, E. Adam, and R. Ismail, 'Detecting Sirex noctilio grey-attacked and lightning-struck pine trees using airborne hyperspectral data, random forest and support vector machines classifiers', *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 88, pp. 48–59, Feb. 2014, doi: 10.1016/j.isprsjprs.2013.11.013.

[319] A. Cruz *et al.*, 'Detection of grapevine yellows symptoms in Vitis vinifera L. with artificial intelligence', *Computers and Electronics in Agriculture*, vol. 157, pp. 63–76, Feb. 2019, doi: 10.1016/j.compag.2018.12.028.

[320] F. Rançon, L. Bombrun, B. Keresztes, and C. Germain, 'Comparison of SIFT Encoded and Deep Learning Features for the Classification and Detection

of Esca Disease in Bordeaux Vineyards', *Remote Sensing*, vol. 11, no. 1, p. 1, Jan. 2019, doi: 10.3390/rs11010001.

[321] M. Kerkech, A. Hafiane, and R. Canals, 'Deep leaning approach with colorimetric spaces and vegetation indices for vine diseases detection in UAV images', *Computers and Electronics in Agriculture*, vol. 155, pp. 237–243, Dec. 2018, doi: 10.1016/j.compag.2018.10.006.

[322] A. Cina, P. Dabove, A. M. Manzino, and M. Piras, 'Network Real Time Kinematic (NRTK) Positioning – Description, Architectures and Performances', in *Satellite Positioning - Methods, Models and Applications*, S. Jin, Ed. InTech, 2015.

[323] L. Tzutalin, 'LabelImg. Git code', 2015. https://github.com/tzutalin/labelImg (accessed May 04, 2020).

[324] TensorFlow, 'Tensorflow Core', *TensorFlow Core*, Apr. 02, 2020. https://www.tensorflow.org/tutorials/load_data/tfrecord?hl=it (accessed May 04, 2020).

[325] F. Pedregosa, G. Varoquaux, Gramfort, and Michel, 'Scikit-learn: Machine Learning in Python', *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[326] K. He, X. Zhang, S. Ren, and J. Sun, 'Deep Residual Learning for Image Recognition', *arXiv:1512.03385 [cs]*, Dec. 2015, Accessed: Jan. 05, 2021. [Online]. Available: http://arxiv.org/abs/1512.03385.

[327] 'Single Camera Calibrator App - MATLAB & Simulink - MathWorks Italia'. https://it.mathworks.com/help/vision/ug/single-camera-calibrator-app.html (accessed Jan. 12, 2021).

[328] J. Brown and AbhimanyuAryan, 'FasterRCNNTutorial', *GitHub*, Jan. 03, 2021. https://github.com/jaspereb/FasterRCNNTutorial (accessed May 07, 2020).

Appendix A

MATLAB routine for HSI splitting in single bands

Code for reading ENVI file as a 3-D Matlab matrix and for HSI splitting in single bands.

```
%‘init.m file’
%initialize matlab and folders

current_folder = pwd;
addpath(genpath(current_folder));
utilities;
algorithms;
save_envi;
disp('Initialization done!');

%’hyper_per_bands.m file’
%inizialize ENVI reader

run init.m

% define the original hypercubes folder
dname='C:\Users\Musci\Desktop\hypercube_per_bands\original\';

filteredDir=strcat(dname,'\*.hdr');% concatenate directory
dList=dir(filteredDir); %list files in the directory
n_file=length(dList);% count files in the directory

% define the destination folder
current_folder=('C:\Users\Musci\Desktop\2019_Sensors\hsi_toolbox.t
ar\hsi_toolbox\output\');

for i=1:n_file

    %read data (.dat) and metadata (.hdr)
    datafolder = erase(strcat(dname,dList(i).name),'.hdr');
    hs = loadenvi(datafolder);
    info = hsinfo(datafolder);
    [h,w,n_bande]=size(hs); %image size

    % define the destination folder
    current_folder=('C:\Users\Musci\Desktop\2019_Sensors\hsi_toolb
    ox.tar\hsi_toolbox\output\band\');
    if i == 1
        for k=1: n_bande
            Banda = hs(:,:,k);
            Band_name= info.wavelength(:,k);
            Band_name_2num= [Band_name{1:end}];

            % define a folder for each band
            mkdir(strcat(current_folder,num2str(Band_name_2num)));
        end
    end
    %split the hypercubes in band and save .tiff file

    for k=1:n_bande
      Banda = hs(:,:,k);
      Band_name= info.wavelength(:,k);
      Band_name_2num= [Band_name{1:end}];
      specificFolder=strcat(current_folder,num2str(Band_name_2num)
      ,'\');
      img_name =(strcat(specificFolder,dList(i).name(32:end-
      14),'_','Band_',num2str(Band_name_2num),'.tif'));

      % Create a new hypercube (.tiff)
      t = Tiff(img_name,'w');

      %tiff metadata
      tagstruct.ImageLength = size(Banda,1);
      tagstruct.ImageWidth = size(Banda,2);
      tagstruct.Compression = Tiff.Compression.None;
      tagstruct.SampleFormat = Tiff.SampleFormat.IEEEFP;
      tagstruct.Photometric = Tiff.Photometric.MinIsBlack;
      tagstruct.BitsPerSample = 64;
      tagstruct.SamplesPerPixel = 1;
      tagstruct.PlanarConfiguration =
Tiff.PlanarConfiguration.Chunky;
      t.setTag(tagstruct);

      write(t,Banda);
      close(t);
    end
end
```

Appendix B

MATLAB routine for HSI camera calibration

Customization of the Matlab *Camera Calibrator* tool [327]. Code for HSI camera calibration.

```
%%insert path
path='C:\Users\Musci\Desktop\2019_Sensors\Matlab_code\hyper_band\;
dList_folder=dir(path);
n_folder= length(dList_folder);
folder_in = (dList_folder(i).name);
filteredDir = strcat(path,folder_in,'\*.tif');

%%create table structure for storing camera parameters and errors
camearaParams_table_out = [];
estimationErrorsIntrinsicsErrors_table_out = [];

for i=3:(n_folder)
    folder_in = (dList_folder(i).name);
    filteredDir = strcat(path,folder_in,'\*.tif');
    dList = dir(filteredDir);

    %clean variabile
    imageFileNames = [];

    for j=1:length(dList)
        image = strcat(path,folder_in,'\', dList(j).name);
        imageFileNames{j}= (image);
    end

%%Detect checkerboards in images
    [imagePoints, boardSize, imagesUsed] =
detectCheckerboardPoints(imageFileNames);
    imageFileNames = imageFileNames(imagesUsed);

%%Read image size from the 1st image
    for j=1:length(dList)
        originalImage = imread(imageFileNames{1});
        [mrows, ncols, ~] = size(originalImage);
    end

%%Generate world coordinates of the corners of the squares
    squareSize = 10;  % in units of 'centimeters'
```

```
      worldPoints = generateCheckerboardPoints(boardSize,
      squareSize);

%%Calibrate the camera
    [cameraParams, imagesUsed, estimationErrors] =
estimateCameraParameters(imagePoints, worldPoints, ...
        'EstimateSkew', true, 'EstimateTangentialDistortion',
true, ...
        'NumRadialDistortionCoefficients', 3, 'WorldUnits',
'centimeters', ...
        'InitialIntrinsicMatrix', [], 'InitialRadialDistortion',
[], ...
        'ImageSize', [mrows, ncols]);

    %% Save figure
%"Mean Reprojection Errors per image"
%     h1=figure; showReprojectionErrors(cameraParams);
%     ReprojectionErrors_folder=
'C:\Users\Musci\Documents\MATLAB\Calibrazione_geometrica_iperspett
rale\Matlab_code\080319_cameraParams\ReprojectionErrors\';
%     filename =
strcat(ReprojectionErrors_folder,folder_in,'_','ReprojectionErrors
');
%     saveas(h1,filename,'png');

%     "Extrinsic Parameter visualization"
%     h2=figure; showExtrinsics(cameraParams, 'CameraCentric');
%     PatternCentric_folder =
'C:\Users\Musci\Documents\MATLAB\Calibrazione_geometrica_iperspett
rale\Matlab_code\080319_cameraParams\PatternCentric\';
%     filename = strcat(PatternCentric,folder_in,'_',' Extrinsic
Parameter');
%     saveas(h2,filename,'png');

%%Display parameter estimation errors
%   displayErrors(estimationErrors, cameraParams);
%   estimationErrors_array = [estimationErrors.IntrinsicsErrors];
%   estimationErrorsIntrinsicsErrors_array =
[estimationErrors.IntrinsicsErrors.SkewError,
estimationErrors.IntrinsicsErrors.FocalLengthError,
estimationErrors.IntrinsicsErrors.PrincipalPointError,
estimationErrors.IntrinsicsErrors.RadialDistortionError,
estimationErrors.IntrinsicsErrors.TangentialDistortionError];
%   estimationErrorsIntrinsicsErrors_table =
array2table(estimationErrorsIntrinsicsErrors_array,
'VariableNames',{'SkewError','FocalLengthErrorX','FocalLengthError
Y','PrincipalPointErrorX','PrincipalPointErrorY','RadialDistortion
ErrorX','RadialDistortionErrorY','RadialDistortionErrorZ','Tangent
ialDistortionErrorX','TangentialDistortionErrorY'});

%Write table "estimationErrorsIntrinsicsErrors_table_out"

%   estimationErrorsIntrinsicsErrors_table_out =
[estimationErrorsIntrinsicsErrors_table_out;
estimationErrorsIntrinsicsErrors_table];

%
writetable(estimationErrorsIntrinsicsErrors_table_out,'C:\Users\Mu
sci\Documents\MATLAB\Calibrazione_geometrica_iperspettrale\Matlab_
code\080319_cameraParams\estimationErrorsIntrinsicsErrors_table_ou
t');
```

```
%%create undistorted image (.tiff)
    [filepath,name,ext] = fileparts(imageFileNames{1});
    undImage = undistortImage(originalImage, cameraParams);
    undImage  = double(undImage);
    undImage_folder =
'C:\Users\Musci\Desktop\2019_Sensors\Matlab_code\test_data\undist\
';
    undName = strcat(undImage_folder,name(33:end-13) ,'_',
folder_in ,'_','undIm.tif');

     t = Tiff(undName,'w');

     %tiff metadata
     tagstruct.ImageLength = size(undImage,1);
     tagstruct.ImageWidth = size(undImage,2);
     tagstruct.Compression = Tiff.Compression.None;
     tagstruct.SampleFormat = Tiff.SampleFormat.IEEEFP;
     tagstruct.Photometric= Tiff.Photometric.MinIsBlack;
     tagstruct.BitsPerSample = 64;
     tagstruct.SamplesPerPixel = 1;
     tagstruct.PlanarConfiguration =
Tiff.PlanarConfiguration.Chunky;
     t.setTag(tagstruct);

     write(t,undImage);
     close(t);

%%create the table of " camera parameters"
name= str2num(folder_in);
camearaParams_array = [cameraParams.RadialDistortion,
cameraParams.TangentialDistortion,cameraParams.FocalLength,
cameraParams.PrincipalPoint, cameraParams.Skew,
cameraParams.MeanReprojectionError,name];
camearaParams_table =
array2table(camearaParams_array,'VariableNames',{'RadialDistortion
_X','RadialDistortion_Y','RadialDistortion_Z','TangentialDistortio
n_X','TangentialDistortion_Y','FocalLength_X','FocalLength_Y',
'PrincipalPoint_X','PrincipalPoint_Y',
'Skew','MeanReprojectionError','name'});
camearaParams_table_out =
[camearaParams_table_out;camearaParams_table];

writetable(camearaParams_table_out,'C:\Users\Musci\Desktop\2019_Se
nsors\Matlab_code\test_data\undist\camera_out');

end
```

Appendix C

MATLAB routine for HSI band co-registration

Customization of the Matlab *imregconfig function* [192]. Code for HSI band co-registration.

```
%insert hypercube name
hypercube = imread('I_HSI_100_Luce2.tif');
nb = size(hypercube,3); %count band number
dim_hyp = size(hypercube);
coreg_hyper = zeros(dim_hyp);
original = hypercube(:,:,1); %select reference band
[optimizer, metric] = imregconfig('monomodal');
optimizer.MaximumIterations = 400;
optimizer.MaximumStepLength = 0.1;
optimizer.MinimumStepLength = 5e-4;
optimizer.GradientMagnitudeTolerance = 5e-5;
coreg_hyper(:,:,1) = original;
for i = 2:nb
    distorted = hypercube(:,:,i);
    movingRegistered = imregister(distorted, original, 'affine',
optimizer, metric);
    coreg_hyper(:,:,i) = movingRegistered;
end
% Creating a new hypercube (.tiff)
t = Tiff('I_HSI_100_Luce2_coR_4.tif','w'); %insert new name
%tiff metadata
tagstruct.ImageLength = size(coreg_hyper, 1);
tagstruct.ImageWidth = size(coreg_hyper, 2);
tagstruct.Compression = Tiff.Compression.None;
tagstruct.SampleFormat = Tiff.SampleFormat.IEEEFP; % image format
tagstruct.Photometric = Tiff.Photometric.MinIsBlack;
tagstruct.ExtraSamples = Tiff.ExtraSamples.Unspecified;
tagstruct.BitsPerSample = 64; %bit number
tagstruct.SamplesPerPixel = 100; %replace with number of bands
tagstruct.PlanarConfiguration = Tiff.PlanarConfiguration.Chunky;
t.setTag(tagstruct);

write(t,coreg_hyper);
close(t);
```

Appendix D

Random Forest implementation in Python

Random Forest algorithm in Python. Code for the semantic segmentation of UAV RGB images.

```
#open the Anaconda environment  and the notebook
(rforest)C:> cd ''directory where this notebook is''
(rforest)C:''directory where this notebook is''> jupyter notebook

#import necessary libraries
import os
import json
import numpy as np
import pandas as pd
import rasterio as rio
from pathlib import Path
from shapely import geometry
from sklearn.metrics import confusion_matrix
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split

#Configurations
test_frac = 0.4
rnd_seed = 37

#Directory
img_dirs = [
    Path("C:/Users/Musci/Desktop/Random_forest/train/"),
    Path("C:/Users/Musci/Desktop/Random_forest/val/")
]
shape_fpaths = [

Path("C:/Users/Musci/Desktop/Random_forest/train/via_region_data.j
son"),

Path("C:/Users/Musci/Desktop/Random_forest/val/via_region_data.jso
n")
]
data_paths = [
    Path("C:/Users/Musci/Desktop/Random_forest/train/data.npy"),
    Path("C:/Users/Musci/Desktop/Random_forest/val/data.npy")]
label_paths = [
Path("C:/Users/Musci/Desktop/Random_forest/train/label_map.json"),

Path("C:/Users/Musci/Desktop/Random_forest/val/label_map.json"),]


#Create image mask
def create_image_mask(a_polygon, class_name, img_arr, label_map):
    assert isinstance(img_arr.shape, tuple) and len(img_arr.shape)
== 2
    if not a_polygon.is_valid:
        a_polygon = a_polygon.buffer(0)
    assert a_polygon.is_valid is True
    try:
        minx, miny, maxx, maxy = a_polygon.bounds
        maxx += 1
        maxy += 1
        r_grid, c_grid = np.meshgrid(np.arange(minx, maxx),
np.arange(miny, maxy), indexing='ij')
        lattice =
geometry.MultiPoint(list(zip(r_grid.ravel().tolist(),
c_grid.ravel().tolist())))
        try:
            points_inside = a_polygon.intersection(lattice)
        except Exception as exc:
            points_inside = None
            print('Exception: ', exc.__class__.__name__)
            print('Polygon: ', a_polygon)
            print('Lattice Bounds: ', lattice.bounds,)
            print('Warning: The <img_arr> will be returned
unaltered')
    except Exception as e:
        points_inside = None
        print('Exception: ', e.__class__.__name__)
        print('Polygon: ', a_polygon)
    if points_inside:
        error_flag = False
        pts = np.array([(p.x, p.y) for p in points_inside],
dtype=int)
        pts_x = pts[:, 0].tolist()
        pts_y = pts[:, 1].tolist()
        pts_idx = (pts_x, pts_y)
        img_arr[pts_idx] = label_map[class_name]
        # img_arr = np.flip(img_arr, axis=0)
    else:
        error_flag = True
    return img_arr, error_flag

#Associate pixel to class label and create a .json file
def process(
    img_dir,
    shape_fpath,
    data_path,
    label_map_path,
    remove_nodata=False,
    test_frac=0.4,
    rnd_seed = 37
):
    aoi_df = pd.DataFrame(columns=('FileName', 'Polygon',
'ClassName'))
    with open(shape_fpath) as json_file:
        annotations = json.load(json_file)
        for fkey in annotations.keys():
            fregions = annotations[fkey]['regions']
            if isinstance(fregions, list):
                labeled_aois = [
                    (
                        fkey,
                        geometry.Polygon(
                            list(
                                zip(

r['shape_attributes']['all_points_y'],

r['shape_attributes']['all_points_x']
                                )
                            )
                        ),

list(r['region_attributes']['name'].keys())[0]
                    )
                    for r in fregions
                ]
                fdf = pd.DataFrame(labeled_aois,
columns=('FileName', 'Polygon', 'ClassName'))
                aoi_df = pd.concat((aoi_df, fdf),
ignore_index=True)

    u_class = set(aoi_df['ClassName'].tolist())
    print(u_class)
    u_files = set(aoi_df['FileName'].tolist())

    label_map = {'nodata': 0}
    l = 1
    for c in u_class:
        label_map[c] = l
        l += 1

    labeled_data = None
    fgroups = aoi_df.groupby(['FileName'])
    for f in u_files:
        g = fgroups.get_group(f)
        plist, clist = g['Polygon'].tolist(),
g['ClassName'].tolist()
        fpath = img_dir / f
        opath = img_dir / (fpath.stem + '_Classified.jpg')
        class_im = None
        with rio.open(fpath) as imptr:
            class_im = np.full(imptr.shape,
fill_value=label_map['nodata'], dtype=np.uint8)
            for ply, lbl in zip(plist, clist):
                class_im, error_flag = create_image_mask(
                    a_polygon=ply, class_name=lbl,
img_arr=class_im, label_map=label_map
                )
                if error_flag:
                    print('Filename: ' + fpath.name +
''.join(fpath.suffixes) + '\n')

#List class label
data_dfs = list()
```

```
for a, b, c, d in zip(img_dirs, shape_fpaths, data_paths,
label_paths):
    print(a, b, c , d)
    data_dfs.append(process(a, b, c, d))


#Annotated images and Label Map directory
udata_path =
Path("C:/Users/Musci/Desktop/Random_forest/Data/Data.npy")
ulabel_path =
Path("C:/Users/Musci/Desktop/Random_forest/Data/Label_Map.json")

#Rectify imbalanced distribution of classes
balanced_data_path =
Path("C:/Users/Musci/Desktop/Random_forest/Data/Balanced_Data.npy"
)
c_ids, freqs = np.unique(ar=labeled_data[:, -1],
return_counts=True)
assert c_ids.size > 1
sample_size = np.min(freqs)
samples = list()
for c in c_ids:
    data_c = labeled_data[labeled_data[:, -1] == c]
    n = data_c.shape[0]
    if n > sample_size:
        sample_c, _ = train_test_split(
            data_c, train_size=sample_size, shuffle=True
        )
    elif n == sample_size:
        sample_c = data_c
    else:
        raise AssertionError("Sample size is bigger than available
data!")
    samples.append(sample_c)
balanced_data = np.concatenate(samples, axis=0)
balanced_data.dump(balanced_data_path)

#Prepare data for training
bmpath = balanced_data_path.absolute().parent /
(balanced_data_path.stem + '.mmap')
bmm = np.memmap(bmpath.stem + '.mmap', dtype=balanced_data.dtype,
mode='w+', shape=balanced_data.shape)
bmm[:] = balanced_data[:]
del balanced_data

mpath = udata_path.absolute().parent / (udata_path.stem + '.mmap')
mm = np.memmap(mpath.stem + '.mmap', dtype=labeled_data.dtype,
mode='w+', shape=labeled_data.shape)
mm[:] = labeled_data[:]
del labeled_data

x = ((bmm[:, :-1]).astype(np.float)) / 255.0
y = (bmm[:, -1]).astype(np.int)
x_train, x_test, y_train, y_test = train_test_split(x, y,
test_size=test_frac, random_state=rnd_seed, shuffle=True)

trainx_fp = udata_path.absolute().parent / 'train_x.npy'
trainy_fp = udata_path.absolute().parent / 'train_y.npy'
testx_fp = udata_path.absolute().parent / 'test_x.npy'
testy_fp = udata_path.absolute().parent / 'test_y.npy'
```

```
x_train.dump(trainx_fp)
y_train.dump(trainy_fp)
x_test.dump(testx_fp)
y_test.dump(testy_fp)

trainx_mm = (udata_path.absolute().parent / 'train_x.mmap').stem +
'.mmap'
trainy_mm = (udata_path.absolute().parent / 'train_y.mmap').stem +
'.mmap'
testx_mm = (udata_path.absolute().parent / 'test_x.mmap').stem +
'.mmap'
testy_mm = (udata_path.absolute().parent / 'test_y.mmap').stem +
'.mmap'

tnx = np.memmap(trainx_mm, dtype=x_train.dtype, mode='w+',
shape=x_train.shape)
tny = np.memmap(trainy_mm, dtype=y_train.dtype, mode='w+',
shape=y_train.shape)
ttx = np.memmap(testx_mm, dtype=x_test.dtype, mode='w+',
shape=x_test.shape)
tty = np.memmap(testy_mm, dtype=y_test.dtype, mode='w+',
shape=y_test.shape)

tnx[:] = x_train[:]
tny[:] = y_train[:]
ttx[:] = x_test[:]
tty[:] = y_test[:]

del x_train
del y_train
del x_test
del y_test

#Train Model
rfc = RandomForestClassifier(
    n_estimators=64,
    max_features="sqrt",
    oob_score=True,
    warm_start=True,
    random_state=rnd_seed,
    n_jobs=4,
    verbose=1
)
# rfc.fit(x_train, y_train)
rfc.fit(tnx, tny)

#Load validation dataset
val_data = np.load(data_paths[1], allow_pickle=True)
assert len(val_data.shape) == 2
vdat = (val_data[:,:-1].astype(np.float)) / 255.0
vcls = val_data[:, -1].astype(np.int)

#Predict
y_pred_test = rfc.predict(ttx)
y_pred_validation = rfc.predict(vdat)

xx, yy = np.unique(y_pred_validation, return_counts=True)
print(xx)
print(yy)
#Validate
error_mat_test = confusion_matrix(tty, y_pred_test)
```

```
error_mat_validation = confusion_matrix(vcls, y_pred_validation)

print(error_mat_test)
print(error_mat_validation)

#Calculate feature_importance
rfc.feature_importances_

#Calculate OOB score
rfc.oob_score_

#Save segmented images
vdir = img_dirs[1]
vdf = data_dfs[1]
ufs = set(vdf['FileName'].tolist())
for f in ufs:
    ipath = vdir / f
    opath = vdir / (ipath.stem + '_Prediction.png')
    cpath = vdir / (ipath.stem + '_Classified.jpg')
    with rio.open(ipath) as ptr:
        xmeta = {
        'driver': 'PNG',
        'dtype': 'uint8',
        'nodata': None,
        'width': ptr.meta['width'],
        'height': ptr.meta['height'],
        'count': 1 ,
        'crs': None,
        'transform': ptr.meta['transform'],
        'interleave': 'band'
    }
        bb = None
        b1 = None
        with rio.open(cpath) as cptr:
            b1 = cptr.read(1).astype(np.uint8)
            bb = np.zeros_like(b1).astype(np.uint8)
        meta = ptr.meta.copy()
        meta['count'] = 1
        arr1 = (ptr.read(1)).ravel()
        arr2 = (ptr.read(2)).ravel()
        arr3 = (ptr.read(3)).ravel()
        arr = np.stack((arr1, arr2, arr3), axis=-1)
        arr = (arr.astype(np.float)) / 255.0
        # arr = np.moveaxis(arr, 0, -1).reshape(-1, *ashp[:1])
        labs = rfc.predict(arr.astype(np.int))
        pred_arr = (labs.reshape(ptr.shape)).astype(np.uint8)
        ba = np.zeros_like(bb).astype(np.uint8)
        p = b1!=0
        q = pred_arr!=0
        ba[np.logical_or(p, q)] = 255
        with rio.open(opath, 'w', **xmeta) as op:
            #op.write(ba, 1)
#             a, b = np.unique(pred_arr, return_counts=True)
#             print(a)
#             print(b)
            op.write(255 * pred_arr, 1)
            #op.write(bb, 3)
            #op.write(b1, 2)
```

Appendix E

Faster R-CNN environment, configuration files and Python code

Installation of software packages and Faster R-CNN configuration. Python-code customization of the Faster R-CNN [328] for the plant disease detection from UAV RGB images.

```
#Step 1: Create an environment.yml file with tensorflow packages
name: Faster R-CNN
channels:
  - anaconda
  - conda-forge
  - numba
  - defaults
dependencies:
  - cudatoolkit=9.0=1
  - sqlite=3.29.0=he774522_0
  - cloudpickle=1.2.2=py_1
  - cycler=0.10.0=py_2
  - cytoolz=0.10.1=py36hfa6e2cd_0
  - dask-core=2.9.0=py_0
  - decorator=4.4.1=py_0
  - freetype=2.10.0=h563cfd7_1
  - kiwisolver=1.1.0=py36he980bc4_0
  - mkl-service=2.3.0=py36hfa6e2cd_0
  - networkx=2.4=py_0
  - olefile=0.46=py_0
  - pyparsing=2.4.5=py_0
  - pyqt=5.9.2=py36h6538335_4
  - python-dateutil=2.8.1=py_0
  - sip=4.19.8=py36h6538335_1000
  - six=1.13.0=py36_0
  - tk=8.6.10=hfa6e2cd_0
  - toolz=0.10.0=py_0
  - tornado=6.0.3=py36hfa6e2cd_0
  - blas=1.0=mkl
  - ca-certificates=2019.11.27=0
  - certifi=2019.11.28=py36_0
  - geos=3.7.1=h33f27b4_0
  - graphviz=2.38=hfd603c8_2
  - hdf5=1.8.20=hac2f561_1
  - icc_rt=2019.0.0=h0cc432a_1
  - icu=58.2=ha66f8fd_1
  - imageio=2.6.1=py36_0
  - intel-openmp=2019.4=245
  - jpeg=9b=hb83a4c4_2
  - libopencv=3.4.2=h20b85fd_0
  - libpng=1.6.37=h2a8f88b_0
  - libtiff=4.0.10=hb898794_2
  - matplotlib=3.1.1=py36hc8f65d3_0
  - mkl=2019.4=245
  - mkl_fft=1.0.15=py36h14836fe_0
  - mkl_random=1.1.0=py36h675688f_0
  - numpy=1.17.3=py36h4ceb530_0
  - numpy-base=1.17.3=py36hc3f5095_0
  - openssl=1.1.1d=he774522_3
  - pillow=6.2.1=py36hdc69c19_0
  - pip=19.3.1=py36_0
  - python=3.6.9=h5500b2f_0
  - pytz=2019.3=py_0
  - pywavelets=1.1.1=py36he774522_0
  - qt=5.9.7=vc14h73c81de_0
  - scikit-image=0.15.0=py36ha925a31_0
  - scipy=1.3.2=py36h29ff71c_0
  - setuptools=42.0.2=py36_0
  - vc=14.1=h0510ff6_4
  - vs2015_runtime=14.16.27012=hf0eaf9b_0
  - wheel=0.33.6=py36_0
  - wincertstore=0.2=py36h7fe50ca_0
  - xz=5.2.4=h2fa13f4_4
  - zlib=1.2.11=h62dcd97_3
  - zstd=1.3.7=h508b16e_0
  - llvmlite=0.28.0=py36h6538335_0
  - pip:
    - absl-py==0.8.1
    - attrs==19.3.0
    - backcall==0.1.0
    - bleach==1.5.0
    - colorama==0.4.1
    - cython==0.29.14
    - dask==2.9.0
    - defusedxml==0.6.0
    - entrypoints==0.3
    - enum34==1.1.6
    - h5py==2.10.0
    - html5lib==0.9999999
    - importlib-metadata==1.2.0
    - ipykernel==5.1.3
    - ipython==7.10.1
    - ipython-genutils==0.2.0
```

```
    - ipywidgets==7.5.1
    - jedi==0.15.1
    - jinja2==2.10.3
    - jsonschema==3.2.0
    - jupyter==1.0.0
    - jupyter-client==5.3.4
    - jupyter-console==6.0.0
    - jupyter-core==4.6.1
    - keras==2.1.5
    - keras-applications==1.0.8
    - keras-preprocessing==1.1.0
    - markdown==3.1.1
    - markupsafe==1.1.1
    - mask-rcnn==2.1
    - mistune==0.8.4
    - more-itertools==8.0.0
    - nbconvert==5.6.1
    - nbformat==4.4.0
    - notebook==6.0.2
    - opencv-python==4.1.2.30
    - pandocfilters==1.4.2
    - parso==0.5.1
    - pickleshare==0.7.5
    - prometheus-client==0.7.1
    - prompt-toolkit==2.0.10
    - protobuf==3.11.1
    - pycocotools==2.0
    - pydot==1.4.1
    - pygments==2.5.2
    - pyrsistent==0.15.6
    - pywin32==227
    - pywinpty==0.5.7
    - pyyaml==5.2
    - pyzmq==18.1.1
    - qtconsole==4.6.0
    - send2trash==1.5.0
    - split-folders==0.3.1
    - tensorflow-estimator==2.0.1
    - tensorflow-gpu==1.5.0
    - tensorflow-tensorboard==1.5.1
    - tensorflow-utils==0.1.0
    - terminado==0.8.3
    - testpath==0.4.4
    - traitlets==4.3.3
    - wcwidth==0.1.7
    - werkzeug==0.16.0
    - widgetsnbextension==3.5.1
    - zipp==0.6.0
prefix: C:\Users\MusciM\AppData\Local\conda\conda\envs\Faster R-CNN

#Step 2: Installing Tensorflow in a defined environment and activate it
conda env create -f environment.yml
conda activate env
#Step3: Creating the train and testing dataset in the same format
as Pascal VOC dataset
+VOCdevkit
    +VOC2012
        +Annotations #.xml label files
        +JPEGImages  #.jpg images
        +ImageSets
                +Main
                       -DiseasePlant_trainval.txt # a list of .jpg
                       file without extensions because the
                       train.py script reads this file for all the
                       images it is supposed to include.
                       -trainval.txt  # copy of the
                       DiseasePlant_trainval.txt

                + trainingconfig.config
               #config based on the Faster R-CNN
               https://github.com/tensorflow/models/tree/master/r
               esearch/object_detection/samples/configs)

#Step4: Fix the metadata in the xml annotation files. Indeed,
LabelImg grabs the folder name when writing the xml files and this
needs to be VOC2012 [328]
for file in $PWD/*.xml
      do sed -i
        's+/home/maria/Documents/tensorflow/VOCdevkit/JPEGImages/+
        /home/maria/Documents/tensorflow/VOCdevkit/VOC2012/JPEGIma
        ges/+g' $file;
      done

#Step5: create an output folder with --set=train option and -
validation option.
python object_detection/dataset_tools/create_pascal_tf_record.py -
-data_dir=/home/maria/Documents/tensorflow/VOCdevkit --
year=VOC2012 --
output_path=/home/maria/Documents/tensorflow/train.record --
label_map_path=/home/maria/Documents/tensorflow/label.pbtxt --
set=train

python object_detection\dataset_tools\create_pascal_tf_record.py -
-data_dir=C:\Users\Mary\Desktop\tAPI_faster\VOCdevkit --
annotations_dir=TAnnotations --
output_path=C:\Users\Mary\Desktop\tAPI_faster\val.record  --
label_map_path=C:\Users\Mary\Desktop\tAPI_faster\label.pbtxt --
set=val

#Step6:convert train and valdation data in TFRecord format.
      #Step6.1: From xml to csv.
      #save xml_to_csv as .py file

      import os
      import glob
      import pandas as pd
      import xml.etree.ElementTree as ET

      def xml_to_csv(path):
          xml_list = []
          for xml_file in glob.glob(path + '/*.xml'):
              tree = ET.parse(xml_file)
              root = tree.getroot()
              for member in root.findall('object'):
                  value = (root.find('filename').text,
                           int(root.find('size')[0].text),
                           int(root.find('size')[1].text),
                           member[0].text,
                           int(member[4][0].text),
                           int(member[4][1].text),
                           int(member[4][2].text),
                           int(member[4][3].text)
                           )
                  xml_list.append(value)
          column_name = ['filename', 'width', 'height', 'class',
      'xmin', 'ymin', 'xmax', 'ymax']
          xml_df = pd.DataFrame(xml_list, columns=column_name)
          return xml_df


      def main():
          for folder in ['TAnnotations']:
              image_path = os.path.join(os.getcwd(),
      ('VOCdevkit/VOC2012/' + folder))
              xml_df = xml_to_csv(image_path)
              xml_df.to_csv(('VOCdevkit/VOC2012/' + folder +
      '_labels.csv'), index=None)
              print('Successfully converted xml to csv.')

      main()

      #run xml_to_csv.py
      python xml_to_csv.py

      #Step6.2: generate TFRecord file
      #save create_pascal_tf_record as.py file

      """
      Usage:
        # In local folder
        # Create train data:
        python create_pascal_tf_record.py --
      csv_input=CSGO_images\train_labels.csv --
      image_dir=CSGO_images\train --
      output_path=CSGO_images\train.record

        # Create test data:
        python create_pascal_tf_record.py --
      csv_input=CSGO_images\test_labels.csv --
      image_dir=CSGO_images\test --
      output_path=CSGO_images\test.record
      """
      from __future__ import division
      from __future__ import print_function
      from __future__ import absolute_import

      import os
      import io
      import pandas as pd
      import tensorflow as tf

      from PIL import Image
      from object_detection.utils import dataset_util
      from collections import namedtuple, OrderedDict
```

```
flags = tf.app.flags
flags.DEFINE_string('csv_input', '', 'Path to the CSV
input')
flags.DEFINE_string('image_dir', '', 'Path to the image
directory')
flags.DEFINE_string('output_path', '', 'Path to output
TFRecord')
FLAGS = flags.FLAGS


# TO-DO replace this with label map
def class_text_to_int(row_label):
    if row_label == 'FD':
        return 1
    else:
        None

def split(df, group):
    data = namedtuple('data', ['filename', 'object'])
    gb = df.groupby(group)
    return [data(filename, gb.get_group(x)) for filename, x
in zip(gb.groups.keys(), gb.groups)]

def create_tf_example(group, path):
    with tf.gfile.GFile(os.path.join(path,
'{}'.format(group.filename)), 'rb') as fid:
        encoded_jpg = fid.read()
    encoded_jpg_io = io.BytesIO(encoded_jpg)
    image = Image.open(encoded_jpg_io)
    width, height = image.size

    filename = group.filename.encode('utf8')
    image_format = b'jpg'
    xmins = []
    xmaxs = []
    ymins = []
    ymaxs = []
    classes_text = []
    classes = []

    for index, row in group.object.iterrows():
        xmins.append(row['xmin'] / width)
        xmaxs.append(row['xmax'] / width)
        ymins.append(row['ymin'] / height)
        ymaxs.append(row['ymax'] / height)
        classes_text.append(row['class'].encode('utf8'))
        classes.append(class_text_to_int(row['class']))

    tf_example =
tf.train.Example(features=tf.train.Features(feature={
        'image/height': dataset_util.int64_feature(height),
        'image/width': dataset_util.int64_feature(width),
        'image/filename':
dataset_util.bytes_feature(filename),
        'image/source_id':
dataset_util.bytes_feature(filename),
        'image/encoded':
dataset_util.bytes_feature(encoded_jpg),
        'image/format':
dataset_util.bytes_feature(image_format),
              'image/object/bbox/xmin':
      dataset_util.float_list_feature(xmins),
              'image/object/bbox/xmax':
      dataset_util.float_list_feature(xmaxs),
              'image/object/bbox/ymin':
      dataset_util.float_list_feature(ymins),
              'image/object/bbox/ymax':
      dataset_util.float_list_feature(ymaxs),
              'image/object/class/text':
      dataset_util.bytes_list_feature(classes_text),
              'image/object/class/label':
      dataset_util.int64_list_feature(classes),
          }))
          return tf_example

      def main(_):
          writer = tf.python_io.TFRecordWriter(FLAGS.output_path)
          path = os.path.join(os.getcwd(), FLAGS.image_dir)
          examples = pd.read_csv(FLAGS.csv_input)
          grouped = split(examples, 'filename')
          for group in grouped:
              tf_example = create_tf_example(group, path)
              writer.write(tf_example.SerializeToString())

          writer.close()
          output_path = os.path.join(os.getcwd(),
      FLAGS.output_path)
          print('Successfully created the TFRecords:
      {}'.format(output_path))

      if __name__ == '__main__':
          tf.app.run()

      #run create_pascal_tf_record.py in
      C:\Users\Mary\cocoapi\models\research\

      #tf_record for Train datatest
      python
      object_detection/dataset_tools/create_pascal_tf_record.py --
      data_dir=C:/Users/Mary/Desktop/tAPI_faster/VOCdevkit --
      annotations_dir=Annotations --
      output_path=C:/Users/Mary/Desktop/tAPI_faster/train.record -
      -
      label_map_path=C:/Users/Mary/Desktop/tAPI_faster/dataset/lab
      el.pbtxt --set=train

      #tf_record for test datatest
      python
      object_detection\dataset_tools\create_pascal_tf_record.py --
      data_dir=C:\Users\Mary\Desktop\tAPI_faster\VOCdevkit --
      annotations_dir=TAnnotations --
      output_path=C:\Users\Mary\Desktop\tAPI_faster\val.record  --
      label_map_path=C:\Users\Mary\Desktop\tAPI_faster\label.pbtxt
      --set=val

#Step7: set the configuration of Faster R-CNN model using the
pipeline.config file

    #Config file (pipeline.config) Faster R-CNN (ResNET-50)

      model {
```

```
faster_rcnn {
  num_classes: 1
  image_resizer {
    keep_aspect_ratio_resizer {
      min_dimension: 1024  #set
      max_dimension: 1024  #set
    }
  }
  feature_extractor {
    type: "faster_rcnn_resnet50"
    first_stage_features_stride: 16
  }
  first_stage_anchor_generator {
    grid_anchor_generator {
      height_stride: 16
      width_stride: 16
      scales: [2, 4, 16, 32, 64] #set
      aspect_ratios: [0.5, 1.0, 2.0] #set
    }
  }
  first_stage_box_predictor_conv_hyperparams {
    op: CONV
    regularizer {
      l2_regularizer {
        weight: 0.0
      }
    }
    initializer {
      truncated_normal_initializer {
        stddev: 0.00999999977648
      }
    }
  }
  first_stage_nms_score_threshold: 0.0
  first_stage_nms_iou_threshold: 0.699999988079
  first_stage_max_proposals: 100 #set
  first_stage_localization_loss_weight: 2.0
  first_stage_objectness_loss_weight: 1.0
  initial_crop_size: 14
  maxpool_kernel_size: 2
  maxpool_stride: 2
  second_stage_box_predictor {
    mask_rcnn_box_predictor {
      fc_hyperparams {
        op: FC
        regularizer {
          l2_regularizer {
            weight: 0.0
          }
        }
        initializer {
          variance_scaling_initializer {
            factor: 1.0
            uniform: true
            mode: FAN_AVG
          }
        }
      }
      use_dropout: false
      dropout_keep_probability: 1.0
    }
    }
    second_stage_post_processing {
      batch_non_max_suppression {
        score_threshold: 0.300000011921
        iou_threshold: 0.600000023842
        max_detections_per_class: 200
        max_total_detections: 200
      }
      score_converter: SOFTMAX
    }
    second_stage_localization_loss_weight: 2.0
    second_stage_classification_loss_weight: 1.0
  }
}
train_config {
  batch_size: 1
  data_augmentation_options {
    random_horizontal_flip {
    }
  }
  optimizer {
    momentum_optimizer {
      learning_rate {
        manual_step_learning_rate {
          initial_learning_rate: 0.000100000014249
          schedule {
            step: 900000
            learning_rate: 1.99999992421e-05
          }
          schedule {
            step: 1200000
            learning_rate: 1.00000010611e-06
          }
        }
      }
      momentum_optimizer_value: 0.899999976158
    }
    use_moving_average: false
  }
  gradient_clipping_by_norm: 10.0

fine_tune_checkpoint:"C:/Users/Mary/Desktop/tAPI_faster/mode
ls/model/model.ckpt"#set
  from_detection_checkpoint: true
  num_steps: 600 #set
}
train_input_reader {
  label_map_path:
"C:/Users/Mary/Desktop/tAPI_faster/label.pbtxt" #set
  tf_record_input_reader {

input_path:"C:/Users/Mary/Desktop/tAPI_faster/pascal.record"
#set
  }
}
eval_config {
  num_examples: 48 #set
  num_visualizations:48 #set
  min_score_threshold: 0.2 #set
  eval_interval_secs: 300 #set
  metrics_set:"coco_detection_metrics" #set
  visualize_groundtruth_boxes: true

visualization_export_dir:"C:/Users/Mary/Desktop/tAPI_faster/
models/model/eval" #set
  use_moving_averages: false
}
eval_input_reader {
  label_map_path:
"C:/Users/Mary/Desktop/tAPI_faster/label.pbtxt" #set
  shuffle: true
  num_readers: 1
  tf_record_input_reader {
    input_path:
"C:/Users/Mary/Desktop/tAPI_faster/val.record" #set

  }
}

#Config file (pipeline.config) Faster R-CNN (ResNET-101)

model {

  faster_rcnn {
    num_classes: 1
    image_resizer {
      keep_aspect_ratio_resizer {

        min_dimension: 1024 #set


        max_dimension: 1024 #set
      }
    }
    feature_extractor {
      type: "faster_rcnn_resnet101"
      first_stage_features_stride: 16
    }
    first_stage_anchor_generator {
      grid_anchor_generator {
        height_stride: 16
        width_stride: 16
        scales:[0.25, 0.5, 1.0, 2.0] #set
        aspect_ratios: [0.5, 1.0, 2.0] #set
      }
    }
    first_stage_box_predictor_conv_hyperparams {
      op: CONV
      regularizer {
        l2_regularizer {
          weight: 0.0
        }
      }
      initializer {
        truncated_normal_initializer {
          stddev: 0.00999999977648
        }
      }
    }
    first_stage_nms_score_threshold: 0.0
    first_stage_nms_iou_threshold: 0.699999988079
    first_stage_max_proposals: 100 #set
    first_stage_localization_loss_weight: 2.0
    first_stage_objectness_loss_weight: 1.0
    initial_crop_size: 14
    maxpool_kernel_size: 2
    maxpool_stride: 2
    second_stage_box_predictor {
      mask_rcnn_box_predictor {
        fc_hyperparams {
          op: FC
          regularizer {
            l2_regularizer {
              weight: 0.0
            }
          }
          initializer {
            variance_scaling_initializer {
              factor: 1.0
              uniform: true
              mode: FAN_AVG
            }
          }
        }
        use_dropout: false
        dropout_keep_probability: 1.0
      }
    }
    second_stage_post_processing {
      batch_non_max_suppression {
        score_threshold: 0.300000011921
        iou_threshold: 0.500000023842
        max_detections_per_class: 300
        max_total_detections: 300
      }
      score_converter: SOFTMAX
    }
    second_stage_localization_loss_weight: 2.0
    second_stage_classification_loss_weight: 1.0
  }
}

train_config {
  batch_size: 1
  data_augmentation_options {
    random_horizontal_flip {
    }
  }
  optimizer {
    momentum_optimizer {
      learning_rate {
        manual_step_learning_rate {
          initial_learning_rate: 0.000100000014249
          schedule {
            step: 500000
            learning_rate: 1.99999992421e-05
          }
          schedule {
            step: 700000
            learning_rate: 1.00000010611e-06
          }
        }
      }
      momentum_optimizer_value: 0.899999976158 #set
     }
     use_moving_average: false
    }
    gradient_clipping_by_norm: 10.0
    fine_tune_checkpoint:
   "C:/Users/Mary/Desktop/tAPI_faster/models/model/pt/model.ckp
   t" #set
    from_detection_checkpoint: true #set
    num_steps: 2000 #set
   }

   train_input_reader {
    label_map_path:  #set
   "C:/Users/Mary/Desktop/tAPI_faster/dataset/label.pbtxt"
    tf_record_input_reader {
     input_path:  #set
   "C:/Users/Mary/Desktop/tAPI_faster/train.record"
    }
   }

   eval_config {
    num_examples: 48 #set
    num_visualizations:48 #set
    min_score_threshold: 0.2 #set
    eval_interval_secs: 300 #set
    metrics_set: "coco_detection_metrics"  #set
    visualize_groundtruth_boxes: true #set

   visualization_export_dir:"C:/Users/Mary/Desktop/tAPI_faster/
   models/model/eval" #set
    use_moving_averages: false
   }
   eval_input_reader {
    label_map_path:  #set
   "C:/Users/Mary/Desktop/tAPI_faster/dataset/label.pbtxt"
    tf_record_input_reader {
     input_path: #set
   "C:/Users/Mary/Desktop/tAPI_faster/val.record"
    }
   }

#Step8: Training and evaluating the accuracy

python object_detection/model_main.py --alsologtostderr --
model_dir=C:/Users/Mary/Desktop/tAPI_faster/models/model/train --
pipeline_config_path=C:/Users/Mary/Desktop/tAPI_faster/models/mode
l/pt/pipeline.config

python object_detection\eval.py --logtostderr  --
pipeline_config_path=C:\Users\Mary\Desktop\tAPI_faster\models\mode
l\pt\faster_resnet50.config --
checkpoint_dir=C:\Users\Mary\Desktop\tAPI_faster\models\model\trai
n\ --
eval_dir=C:\Users\Mary\Desktop\tAPI_faster\models\model\eval\

#Step9: Visualization of results on Tensorboard
```

```
tensorboard --
logdir=C:/Users/Mary/Desktop/tAPI_faster/models/model --port=8008

#Step10: export_inference_graph

python object_detection\export_inference_graph.py --
input_type=image_tensor --
pipeline_config_path=C:\Users\Mary\Desktop\tAPI_faster\models\mode
l\pt\faster_resnet50.config  --
trained_checkpoint_prefix=C:\Users\Mary\Desktop\tAPI_faster\models
\model\train\model.ckpt-60 --
output_directory=C:\Users\Mary\Desktop\tAPI_faster\models\model\IG
```