JYU DISSERTATIONS 608

Samuli Rahkonen

Deep Learning Architectures for Hyperspectral Imaging Applications

UNIVERSITY OF JYVÄSKYLÄ
FACULTY OF INFORMATION
TECHNOLOGY

JYU DISSERTATIONS 608

Samuli Rahkonen

Deep Learning Architectures for Hyperspectral Imaging Applications

Esitetään Jyväskylän yliopiston informaatioteknologian tiedekunnan suostumuksella
julkisesti tarkastettavaksi Agoran auditoriossa 3
maaliskuun 10. päivänä 2023 kello 12.

Academic dissertation to be publicly discussed, by permission of
the Faculty of Information Technology of the University of Jyväskylä,
in building Agora, Auditorium 3, on March 10, 2023, at 12 o'clock.

JYVÄSKYLÄN YLIOPISTO
UNIVERSITY OF JYVÄSKYLÄ

JYVÄSKYLÄ 2023

Editors
Marja-Leena Rantalainen
Faculty of Information Technology, University of Jyväskylä
Päivi Vuorio
Open Science Centre, University of Jyväskylä

Copyright © 2023, by the author and University of Jyväskylä

ISBN 978-951-39-9299-6 (PDF)
URN:ISBN:978-951-39-9299-6
ISSN 2489-9003

Permanent link to this publication: http://urn.fi/URN:ISBN:978-951-39-9299-6

ABSTRACT

Rahkonen, Samuli
Deep learning architectures for hyperspectral imaging applications
Jyväskylä: University of Jyväskylä, 2023, 44 p. (+included articles)
(JYU Dissertations
ISSN 2489-9003; 608)
ISBN 978-951-39-9299-6 (PDF)

A typical consumer camera captures three bands of light corresponding to red, green and blue colors. A hyperspectral imager captures dozens or even hundreds of bands. A depth sensing camera captures the distance to the target at each pixel. Imaging spectra and depth opens new possibilities for extracting information about the target, and these kind of imagers have already been used in applications in agriculture, astronomy, forestry, medical imaging and other industries. The captured high-dimensional data volumes are large, and extracting meaningful information from them requires advanced and efficient processing methods. Previously, the need for expert manual work has limited the utilization of data in large scale. This research introduces neural network models for solving these problems in a few case applications. It also demonstrates hyperspectral measurement methods: one for radiance approximation and another for angular reflectance measurement by combining a depth camera with a hyperspectral camera.

Keywords: hyperspectral imaging, neural networks, depth sensing, reflectance

TIIVISTELMÄ (ABSTRACT IN FINNISH)

Rahkonen, Samuli
Syväoppivat arkkitehtuurit hyperspektrikuvantamissovelluksille
Jyväskylä: University of Jyväskylä, 2023, 44 s. (+artikkelit)
(JYU Dissertations
ISSN 2489-9003; 608)
ISBN 978-951-39-9299-6 (PDF)

Tavallinen kuluttajakamera kuvaa kolme valon aallonpituuskaistaa: punaisen, vihreän ja sinisen. Hyperspektrikamera kuvaa kymmeniä tai jopa satoja kaistoja. Syvyyskamera taltioi kohteen etäisyyden kuvan jokaista pikseliä kohden. Spektrien ja syvyystiedon yhdistäminen mahdollistaa uuden tiedon tuottamisen kuvattavasta kohteesta. Vastaavia kuvantamislaitteita käytetään ennestään maa- ja metsätaloudessa, tähtitieteessä, lääketieteellisessä kuvantamisessa sekä teollisuuden aloilla. Kuvantamisessa syntyvät suuret moniulotteiset tietomassat vaativat kehittyneitä ja tehokkaita käsittelymenetelmiä. Aiemmin tarve käsin tehtävälle asiantuntijatyölle on rajoittanut tiedon hyödyntämistä suuressa mittakaavassa. Tämä tutkimus esittelee neuroverkkomalleja muutamalle sovellukselle näiden ongelmien ratkaisemiseen. Tutkimus havainnollistaa myös mittausmenetelmiä radianssin likimääräiselle arvioinnille ja kulmariippuvaisen reflektanssin mittaamiselle yhdistämällä syvyys- ja hyperspektrikameroiden havaintoja.

Avainsanat: hyperspektrikuvantaminen, neuroverkot, syvyyskamera, reflektanssi

Author

Samuli Rahkonen
Faculty of Information Technology
University of Jyväskylä
Finland

Supervisors

Associate Professor Ilkka Pölönen
Faculty of Information Technology
University of Jyväskylä
Finland

Docent Sami Äyrämö
Faculty of Information Technology
University of Jyväskylä
Finland

Professor Lauri Kettunen
Faculty of Information Technology
University of Jyväskylä
Finland

Reviewers

Professor Paul Scheunders
Imec-Visionlab
University of Antwerp
Belgium

Associate Professor Tuomas Eerola
School of Engineering Science
Lappeenranta–Lahti University of Technology LUT
Finland

Opponent

Professor Jussi Tohka
Faculty of Health Sciences
University of Eastern Finland
Finland

ACKNOWLEDGEMENTS

First of all, I want to thank my supervisor, Ilkka Pölönen, for welcoming me in to his research group in 2016. I don't think I could have finished this dissertation without his gracious and patient guidance over the years. Thank you to my other supervisors, Sami Äyrämö and Lauri Kettunen. Thank you, Kaisa Miettinen and Pekka Neittaanmäki, for taking me in and letting me work at the university the first time many years ago and giving me the opportunity to see what research work in a university can be like. I also want to thank my financiers: Business Finland, Regional Council of Central Finland, Jane and Aatos Erkko Foundation and University of Jyväskylä.

Thanks to my colleagues and friends Sampsa Kiiskinen, Leevi Annala, Leevi Lind, Anna-Maria Raita-Hakola, Matti Eskelinen, Kimmo Riihiaho, Pauliina Salmi, Jarno Kiesiläinen, Jarno Pasonen, Vilho Halonen and everyone with whom I had the joy to work, solve problems, come up with new ideas and have fun with. I'm going to remember Anna-Maria's wacky playtime activities and unicorns. Thanks to Sampsa Kiiskinen for letting me steal your time over the years. You all deserve a catfish ( ).

I want to thank my family for the upbringing and all the loving support I have been given. Thank you mom for planting the seed of never-ending curiosity of the world and science in me. Thank you dad for being the supportive figure in my life. Thank you to my brother and his infectious interest in natural sciences, which has had a profound effect on me ever since my childhood.

Tampere, 8.12.2022
Samuli Rahkonen

LIST OF ACRONYMS

CNN	Convolutional neural network
FPI	Fabry-Pérot interferometer
GPU	Graphics processing unit
HSI	Hyperspectral imaging
RGB	Red, green, blue
SfM	Structure from motion
ToF	Time-of-flight
UAV	Unmanned aerial vehicle

LIST OF FIGURES

FIGURE 1 A hyperspectral data cube with spatial and spectral dimensions. 14
FIGURE 2 Structure of the included articles towards the research goals. 15
FIGURE 3 The experimental FPI HSI camera in its enclosure. 18
FIGURE 4 The working principle of a Fabry-Pérot interferometer camera. . 19
FIGURE 5 An example RGB image and ToF depth map........................... 20
FIGURE 6 The working principle of time-of-flight depth sensing camera... 21
FIGURE 7 An example of a dense feedforward neural network................ 24
FIGURE 8 The output h of one neuron in a dense feedforward network layer. 25
FIGURE 9 2D convolution operation over image $\mathbf{I}$ using a 3×3 convolution kernel $\mathbf{K}$. 26
FIGURE 10 An example convolutional neural network with dense hidden layers.. 27
FIGURE 11 An example of dense encoder-decoder architecture. 28

CONTENTS

ABSTRACT
TIIVISTELMÄ (ABSTRACT IN FINNISH)
ACKNOWLEDGEMENTS
LIST OF ACRONYMS
LIST OF FIGURES
CONTENTS
LIST OF INCLUDED ARTICLES

1 INTRODUCTION 13
1.1 Background 13
1.2 Objectives and scope 15
1.3 Dissertation structure 16

2 THEORETICAL FOUNDATION 17
2.1 Hyperspectral imaging 17
2.2 Depth sensing 19
2.3 Sensor fusion of hyperspectral and depth information 21
2.4 Neural networks 23
2.4.1 Multilayer feedforward network 23
2.4.2 Convolutional neural network 25
2.4.3 Image segmentation 27
2.4.4 CNN design for hyperspectral data 29

3 RESEARCH RESULTS AND RESEARCH CONTRIBUTION 31
3.1 PI: Convolutional neural networks in skin cancer detection using spatial and spectral domain 31
3.2 PII: Tree species identification using 3D spectral data and 3D convolutional neural network 32
3.3 PIII: Multilabel segmentation of cancer cell culture on vascular structures with deep neural networks 33
3.4 PIV: Method for radiance approximation of hyperspectral data using deep neural network 34
3.5 PV: Reflectance Measurement Method Based on Sensor Fusion of Frame-based Hyperspectral Imager and Time-of-flight Depth Camera 35

4 CONCLUSIONS 38

YHTEENVETO (SUMMARY IN FINNISH) 40

REFERENCES 41

INCLUDED ARTICLES

LIST OF INCLUDED ARTICLES

PI Ilkka Pölönen, **Samuli Rahkonen**, Leevi Annala and Noora Neittaanmäki. Convolutional neural networks in skin cancer detection using spatial and spectral domain. *Proceedings of SPIE Volume 10851: Photonics in Dermatology and Plastic Surgery, SPIE, The International Society for Optical Engineering*, 2019.

PII Ilkka Pölönen, Leevi Annala, **Samuli Rahkonen**, Olli Nevalainen, Eija Honkavaara, Sakari Tuominen, Niko Viljanen and Teemu Hakala. Tree species identification using 3D spectral data and 3D convolutional neural network. *WHISPERS 2018 : 9th Workshop on Hyperspectral Image and Signal Processing, Evolution in Remote Sensing*, 2019.

PIII **Samuli Rahkonen**, Emilia Koskinen, Ilkka Pölönen, Tuula Heinonen, Timo Ylikomi, Sami Äyrämö and Matti A. Eskelinen. Multilabel segmentation of cancer cell culture on vascular structures with deep neural networks. *Journal of Medical Imaging, SPIE*, 2020.

PIV **Samuli Rahkonen** and Ilkka Pölönen. Method for radiance approximation of hyperspectral data using deep neural network. *Impact of scientific computing on science and society, Springer*, pending publication, 2023.

PV **Samuli Rahkonen**, Leevi Lind, Anna-Maria Raita-Hakola, Sampsa Kiiskinen and Ilkka Pölönen. Reflectance measurement method based on sensor fusion of frame-based hyperspectral imager and time-of-flight depth camera. *MDPI Sensors, Sensors and Robotics, Kinect Sensor and Its Application*, 2022.

1 INTRODUCTION

1.1 Background

A typical consumer camera captures three bands of light corresponding to red, green and blue colors. A hyperspectral imager captures dozens or even hundreds of bands. Applications in agriculture, astronomy, forestry, medical imaging and other industries make use of spectral imagers. For example, they have been used in skin cancer diagnosis to detect malignant melanoma (Raita-Hakola et al. 2022), in perishable grocery quality control (Lu et al. 2020) and to preserve our cultural heritage, such as art and historical artifacts (Striova et al. 2020; Bayarri et al. 2019; Sandak et al. 2021).

Another type of imager is a depth sensing camera. Instead of capturing the color of a pixel, a depth camera captures the distance to the target at the pixel. Example applications include robots that need to gather information, detect and plan actions autonomously based on their sensory inputs. Fusing different image sources and extracting meaningful information impose challenging problems.

A hyperspectral camera captures a hyperspectral data cube with spatial and spectral dimensions. The data cube is arranged in a uniform grid where each pixel corresponds to a mixed radiance spectrum of light, ranging from visible light to infrared wavelengths, depending on the application and camera. Figure 1 shows an example hyperspectral cube. In this research, a Fabry-Pérot interferometer (FPI) with a machine vision camera was used.

A quantity commonly measured with spectral imagers is the spectral reflectance of a material, defined as the ratio of reflected and incident light per measured wavelength band. A material with reflectance of one will reflect all radiation incident on it, and a material with reflectance of zero reflects nothing. Respectively, spectral reflectance is a set of reflectances each corresponding to a wavelength channel. Reflectance depends on the directions of incident and reflected light. (Hapke 2012)

Processing hyperspectral data cubes is methodologically, computationally and memory-wise demanding. A large data volume, with potentially hundreds

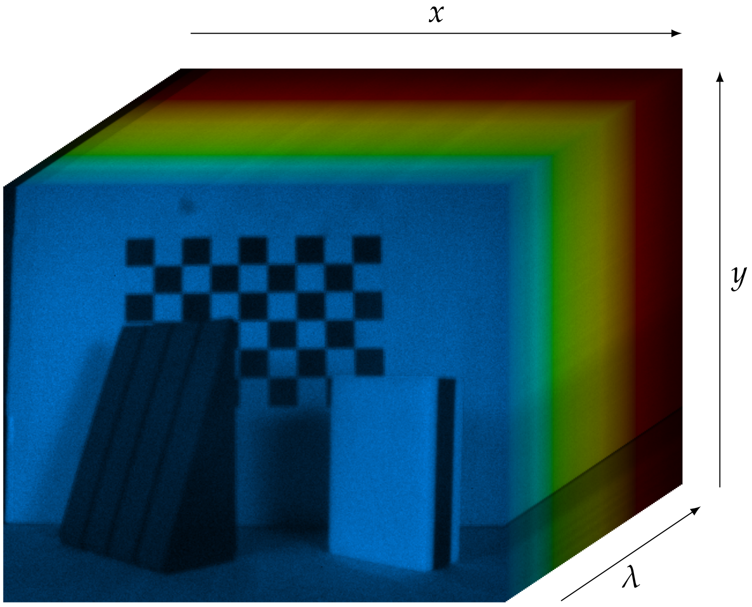


FIGURE 1 A hyperspectral data cube with spatial and spectral dimensions. Each pixel corresponds to a spectrum of the imaged target.

of floating point intensity values per coordinate pixel, requires scalable and parallelizable computation methods. Previously, the demand of expert manual labeling work has limited the utilization of data in large scale. Also, in real world scenes, the pixels′ radiance spectra include mixed light of different compounds with different abundances. Distinguishing these mixed spectra and making decisions based on them requires advanced processing methodologies.

Hyperspectral imaging covers, but is not limited to the following problems:

- Unmixing techniques for determining the composition of an imaging target (Puupponen 2014), such as vertex component analysis (Nascimento and Dias 2005). Unmixing is able to divide spectra to idealized classes of spectral components, typically called endmembers.

- Physical parameter retrieval. For example, skin thickness estimation from medical images, can be achieved by using stochastic models (Annala and Pölönen 2022). In aerial photos, different properties of land could be inferred, such as surface temperature, biomass and leaf area coverage (Bioucas-Dias et al. 2013).

- Image- and pixel-level classification in, for example, aerial photos (Audebert et al. 2019) and medical images (Raita-Hakola et al. 2022). Neural networks have been used for these tasks. They are able to learn distinguishing features from the given data and utilize them to make predictions on previously unseen data.

Research papers of this dissertation consider mainly deep learning–based methodologies with hyperspectral data.

1.2 Objectives and scope

The aim of this dissertation is to find out how deep learning can be used to infer and derive useful information from hyperspectral data and how can spectral data be augmented with depth sensing imaging modalities? The research was done in the fields of medicine, forestry, spectral imaging and depth sensing imaging. Figure 2 depicts the order and structure of the included research papers and how they answer the research questions.

The author starts by defining a convolutional deep learning architecture for hyperspectral data for skin cancer classification and segmentation in Paper PI. A similar convolutional neural network (CNN) was then used in tree species identification with 3D spatial and spectral data in Paper PII. Image segmenting data is further approached in the context of cell culture microscope images in Paper PIII. The experimented techniques were then used as a foundation in approximating radiance from raw hyperspectral data with a CNN in Paper PIV. In Paper PV, the author applied sensor fusion to augment hyperspectral data with depth data from a mid-range depth sensor camera. Data fusion of long distance data was first applied in Paper PII.

The research questions are the following:

Q1. How can we build a deep learning architecture for hyperspectral data?

(a) How to define a deep learning architecture for skin cancer classification?

(b) How to define a deep learning architecture for tree identification?

(c) How to define a deep learning architecture for hyperspectral radiance approximation?

Q2 How well does the CNN work in multilabel cell culture segmentation?

Q3 How can we fuse hyperspectral data with depth information?

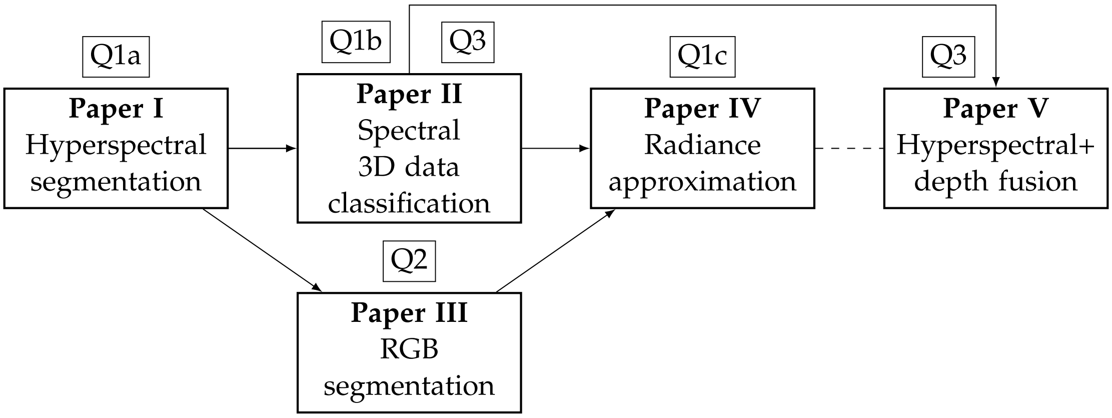


FIGURE 2 Structure of the included articles towards the research goals. The dependencies and influences between papers are delineated with arrows.

1.3 Dissertation structure

The dissertation is structured as the following: Chapter 1 puts the dissertation in a larger context and introduces the research problems. Chapter 2 describes the theoretical foundations. Chapter 3 presents the author contributions, the results, and discussion of each included research article. Chapter 4 summarizes the results and provides the concluding remarks.

2 THEORETICAL FOUNDATION

2.1 Hyperspectral imaging

Hyperspectral imaging (HSI) considers capturing images with specialized hyperspectral cameras. Each image pixel in the hyperspectral cube captures a spectrum of light, and each wavelength is captured with a narrow bandwidth. The spectral and spatial dimensions together can be used to characterize and identify points of interest in the image. (Lillesand et al. 2007)

Compared to regular RGB cameras, an HSI camera captures a data cube over a continuous spectral range. HSI should be separated from multispectral imaging, which refers to only capturing selected noncontinuous wavelength bands with relatively wider wavelength bands (Chang 2007). HSI cameras can be divided into four scanning types: spatial scanners, spectral scanners, snapshot imagers and spatiospectral scanners. The hyperspectral camera used in this dissertation was a spectral scanner.

A spatial scanner requires the imaged object or the camera to be moved for creating a hyperspectral data cube. The scanner sensor captures a slit spectrum as a 2D data array (x, λ). Moving the camera sensor in relation to the target and taking consecutive measurements form a full (x, y, λ) data cube. Here x, y and z refer to spatial dimensions, and λ to the spectral domain. This kind of 2D sensor is called a push broom type scanner. They are often used in remote sensing applications and imaging setups where the target movement is predictable, such as a conveyor belt. Some cameras can also move the sensor behind the optics, enabling scanning without moving the camera. A whisk broom scanner is another special spatial scanner type; it uses a 1D or a narrow 2D sensor. The scanner captures an image by sweeping the imaging area with a single or a few detectors, which requires the scanner to be moved to cover the imaging area. (Chang 2007)

Spectral scanners are frame-based imagers which capture the whole bands and stack them to form a full HSI data cube. The process involves changing the narrow band-pass filter between frames and capturing images on a 2D sensor with different wavelengths (Chang 2007). An adjustable band-pass filter makes

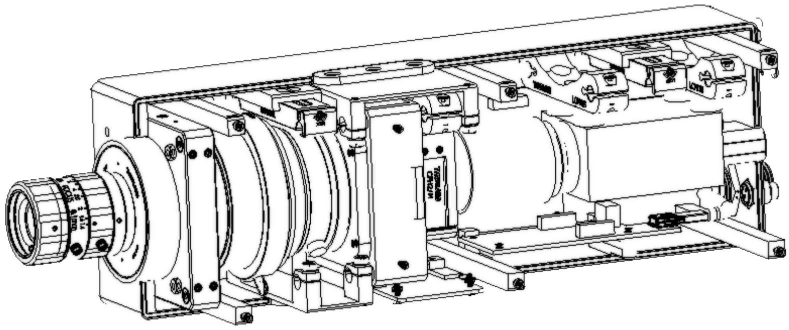


FIGURE 3 The experimental FPI HSI camera in its enclosure. From left to right: optics, a FPI module, optics and a machine vision sensor. Image by Hans Toivanen, Technical Research Centre of Finland Ltd (VTT).

this more practical, such as a piezo-actuated Fabry-Pérot interferometer (Saari et al. 2013). The benefits of this kind of setup are that the wavelengths can be selected through configuration of the filter settings and spatial dimensions are not affected by motion of the camera or target object. However, motion would affect scenes at different wavelengths, as all the bands are not captured simultaneously.

A snapshot hyperspectral imager is able to yield the full hyperspectral data cube at once. The benefits of this kind of method is the short acquisition time, which enables recording even video to capture transient events (Bodkin et al. 2009; Hauser et al. 2017; Toivonen et al. 2020). Depending on the technique, it can be notoriously more difficult computation-wise, as in the case of a computed tomography imaging spectrometer (CTIS). CTIS considers capturing a diffraction pattern on the sensor and reconstructing the hyperspectral data cube from its heavily overlapping spectral information (Okamoto and Yamaguchi 1991).

The spatiospectral scanning technique captures a temporal sequence of spectrally coded images of a scene. It captures consecutive slit spectra to reconstruct an HSI data cube from diagonal slices of the cube. A slit can be regarded as a series of pinholes, and a grating diffracts the rays into wavelength-specific directions. Each projection forms a linear rainbow-colored slice for a spectroscopic image which together form an HSI data cube. (Grusche 2014)

Fabry-Pérot interferometer camera

A Fabry-Pérot interferometer camera is a special type of a spectral scanner capable of capturing complete band frames at a time and building a hyperspectral data cube from them. The camera used in this research, depicted in Figure 3, was developed by Technical Research Centre of Finland Ltd (VTT) (Saari et al. 2013). Figure 4 shows the basic structure of the camera. It is an assembly of optics, an interferometer, filters and a machine vision sensor with an RGB sensor. It captures a hyperspectral data cube that has (x, y) spatial dimensions and a spectral domain.

The camera works by capturing multiple images and varying the interfer-

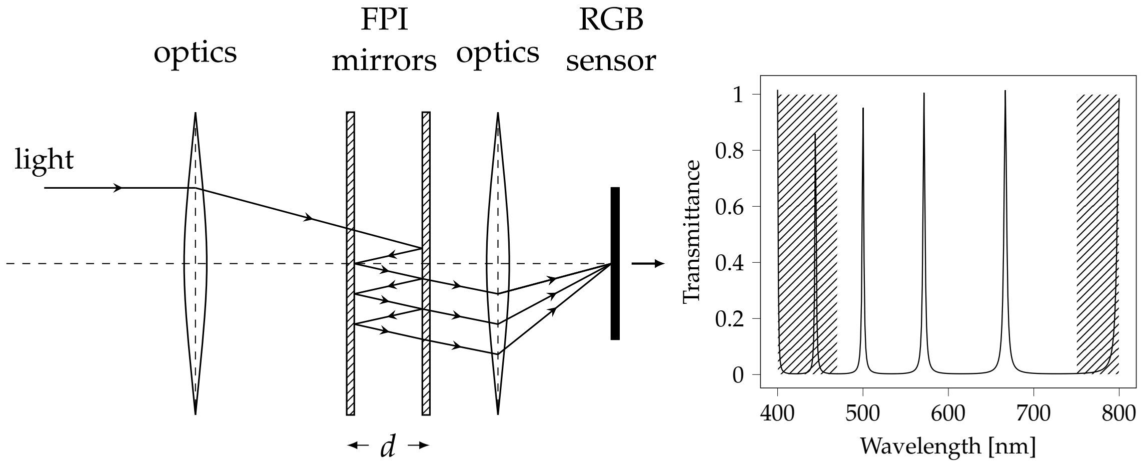


FIGURE 4 The working principle of a Fabry-Pérot interferometer camera: (on the left) light passing through an assembly of optics and semi-transparent mirrors before reaching an RGB sensor. (On the right) three integer multiples of wavelengths of light are passed through the system. The electronically adjustable mirror separation *d* with high- and low-pass filters control which narrow bandwidths are let through.

ometer settings between exposures. The piezo-actuated interferometer consists of two parallel metallic half-mirrors whose separation can be controlled. A beam of light entering the system interferes with itself as it reflects off the mirrors. Only integer multiples of certain wavelengths transmit through the mirrors. The camera uses high- and low-pass filters to block unwanted wavelengths of light, making the setup a narrow-band wavelength filter by controlling the mirror separation. (Eskelinen 2019; Saari et al. 2009)

Conversion to radiance data cube is traditionally done as follows: the raw Bayer matrix RGB sensor data is bilinearly interpolated to produce an array of RGB images. Each image is converted to multiple narrow wavelength radiance bands depending on the number of received peaks on the sensor. On the right in Figure 4, the example plot shows received peaks on the sensor. The coefficients for calculating the radiance for each corresponding wavelength are solved during the camera calibration. The resulting radiance data cube may include larger number of radiance bands than in the given raw data cube. (Eskelinen 2019; Saari et al. 2009)

2.2 Depth sensing

Range imaging and depth sensing refer to imaging techniques that produce a 2D image where every pixel corresponds to the distance to a point in a scene from the perspective of a viewer. The 2D image captured by a depth camera is usually called a range image or a depth map. The pixel values are associated with the

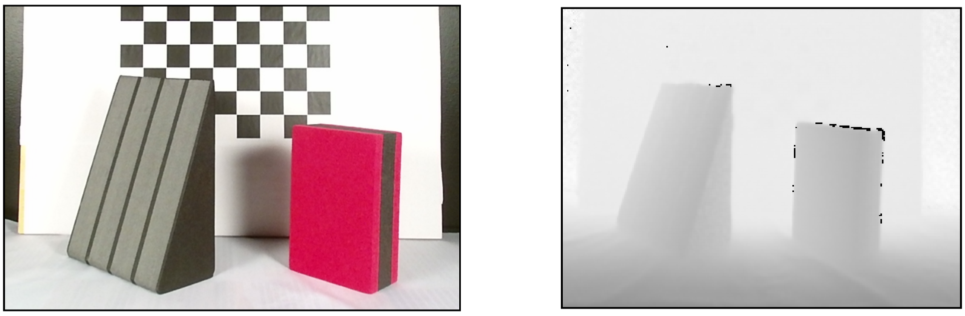


FIGURE 5 (On the left) an RGB image and (on the right) a depth map captured with a ToF camera before image rectification. Lighter pixel color means that the corresponding point in scene space is farther from the camera.

captured surfaces and can be given in physical units, such as meters, if calibrated properly. Figure 5 shows an example depth map. A 2D image is taken from the perspective of the camera, which has to be taken into account when using it together with other imaging modalities. Two example methods for producing depth maps from real-life targets are stereophotogrammetry and time-of-flight (ToF) techniques. (Gokturk et al. 2004)

Another common approach for representing depth information is a point cloud. It stores (x, y, z) locations and does not restrict the spatial information to the uniform camera grid of the camera sensor from a certain viewpoint. Point clouds are a practical for, e.g., fusing point data from multiple sources, allowing representation, analysis and visualization of objects that could otherwise be occluded by other objects in individual camera views.

Stereophotogrammetry refers to using a 2D multi-camera system for imaging a scene and finding pixel-wise distances. The system works by finding the corresponding points (or features) in the images and analyzing the triangles constructed by the projection rays of the two cameras. The intersection of the rays from the different camera locations defines the location of the target. (Gokturk et al. 2004)

In a structure from motion (SfM) model 3D information is inferred from overlapping images without knowing the exact camera locations and orientations. The system is able to compute the location of a moving point or with a moving camera. It is based on finding the image correspondences and solving the camera poses and parameters to create sparse point clouds. Point clouds from a large number of overlapping images can then be used to create a dense point cloud. Aerial photographs are a common use case for this type of photogrammetry, as SfM is used for creating orthophotographs. They are accurate top-down representations of the Earth's surface without camera tilt, with color and distance information. (Iglhaut et al. 2019)

Time-of-flight cameras

ToF systems have been used in radar and Lidar systems for decades to perceive objects in the surrounding world. The basic working principle involves trans-

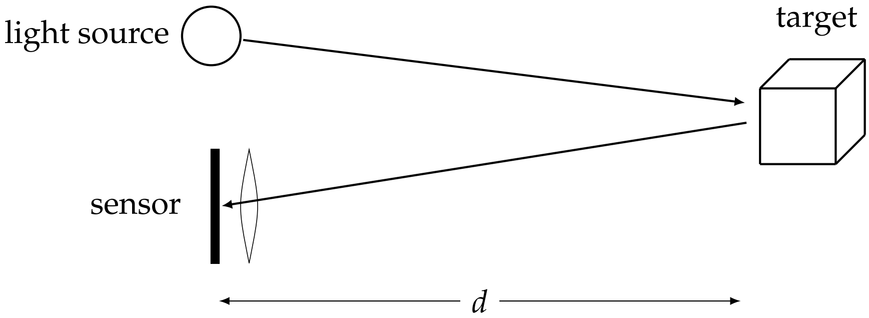


FIGURE 6 The working principle of time-of-flight depth sensing camera. A rapidly pulsating light source illuminates the target object, and the sensor detects the reflecting light. The distance d to the target is calculated from the time difference between the transmitted and received light.

mitting a signal and measuring the time of flight of the returned signal from the target object. The distance is calculated via multiplication of the time of flight and the velocity of the signal. (Gokturk et al. 2004)

ToF cameras measure the distance between the camera and the object of interest in each point in the image by measuring the round trip time of the emitted light pulse, such as infrared light. Figure 6 illustrates the working mechanism. The time of flight is measured from the phase difference of the transmitted and received pulses (Gokturk et al. 2004; Hansard et al. 2012). Error sources have to be taken into account when designing an application around a ToF camera. These include systematic errors, such as noise and data ambiguity, as the sampling frequency restricts the optimal operational range of the measurement. The color, reflectivity and shape of the imaged object affect the reflected infrared light, which shows as amplitude and phase variations on the sensor, the quantities that ToF depth estimation is based on. Other error sources are the non-systematic errors, such as scattering and motion blur (Hansard et al. 2012).

The ToF camera used in this research was a Microsoft Kinect V2 (Sell and O'Connor 2014). Figure 5 depicts cropped frames from its RGB sensor and a depth map constructed with its ToF system.

2.3 Sensor fusion of hyperspectral and depth information

Combining different imaging modalities is challenging. Image sensor fusion requires knowing the optical properties of the sensors, selecting the right optics and finding the sensors' mutual reference frame through calibration. The camera calibration method depends on the type of HSI and depth cameras. Some methods used in RGB image rectification are applicable, such as spatial feature (Hansard et al. 2012) and phase correlation–based methods (Henke et al. 2018; Sun et al. 2019). The resulting 4D hyperspectral point clouds (x, y, z, λ) include spatial dimensions and a spectral dimension.

In multi-camera system calibration, the intrinsic and extrinsic camera parameters are inferred through calibration. Intrinsic parameters describe the internal camera parameters, such as focal length, principal point and skew of the pinpoint camera model. Extrinsic parameters relate the positions and orientations of two cameras to each other i.e., the translation and rotation between camera viewpoints. A common reference frame can be, for example, a series of images of checkerboard pattern, imaged from different angles (Hansard et al. 2012). The tile corners can be automatically detected, and the matching checkerboard grids from the two sets of images are used for estimating the transformation between the two cameras.

Another approach of evaluating the camera's extrinsic parameters is using the principle of Fourier transform and phase correlation. The translation, rotation matrix, and scaling coefficient between HSI images and depth maps are determined based on the Fourier spectra of the calibration images. In theory, the cross-power spectrum between two structurally identical images with relative displacement, scaling or rotation can be described as phase-shifts of the Fourier transforms in the frequency domain. Selecting the maximum peak of the phase correlation between two images taken by two cameras gives the relative translation, scaling or rotation. The benefit of this approach is that it enables automatic calibration of the imaging system. (Henke et al. 2018; Sun et al. 2019)

A depth map captured with a ToF camera can be turned into a point cloud with global coordinates, once we know the intrinsic camera parameters (Szeliski 2022). The estimated global point coordinates would then be transformed to the viewpoint of a hyperspectral camera, using the estimated extrinsic parameters, and projected onto its camera plane. Sensor fusion would then consider matching the projected points to the pixels on the hyperspectral camera plane to form a spectral point cloud. The more detailed sensor fusion of FPI HSI and ToF depth data is described in Paper PV.

Depth can also be inferred from fixed targets with just a single frame-based camera by moving the camera and capturing multiple frames of the target from different locations and orientations. Spectral registration of data cubes is important with a Fabry-Pérot interferometer HSI camera on a moving platform, as frames are captured in a sequence to form a full HSI data cube. HSI data cube bands should be well aligned in the spectral dimension. (Rosnell and Honkavaara 2012)

In the case of long-range UAV (unmanned aerial vehicle) aerial photos, photogrammetrical processing of a series of images can be done as with RGB images, but for each spectral band separately. The geometrical processing of UAV images includes determination of the locations and orientations of the images by using an on-board IMU (inertial measurement unit), GPS/GNSS and adjacent images with corresponding tie points and reference ground control points to form hyperspectral point clouds. The point clouds can be registered together to form a dense point cloud and to further create an orthophotographic mosaic covering large geographical areas. (Rosnell and Honkavaara 2012)

2.4 Neural networks

Neural networks aim to mimic biological neural networks which approximate functions that depend on a large number of inputs. In machine learning, neural networks are used to present a model which can make predictions, classify or generate previously unseen data. They are used in tasks, such as computer vision and speech recognition, where problem solving is often unfeasible using a fixed rule set. In this research, the author used neural networks to infer information from hyperspectral, RGB and depth data.

Neural networks are formed of a large number interconnected neurons i.e., nodes. A typical node multiplies its input values with some learnt weight and passes their sum to an activation function that returns a value. Typical activation functions are linear, logistic sigmoid, rectified linear units and softmax, depending on the neuron's purpose in the network.

A neuron output y is defined as follows:

$$y = g\left(\sum_{i=1}^{n} w_i x_i + b\right) \tag{1}$$

where x is an input vector of length n, w_i are weights, b is bias and $g(x)$ is an activation function. (Goodfellow et al. 2016)

The layout of nodes depends on the architecture of the network (Haykin 1999). Traditionally, neural networks are divided into three classes of network architectures: Single-Layer Feedforward Networks, Multilayer Feedforward Networks and Recurrent Networks (Haykin 1999). A CNN is a special type of neural network which includes at least one convolution operation. A deep neural network is a multilayer neural network with a large number of node layers. Recurrent neural networks have at least one feedback loop where a node's output value is fed back to one of the network nodes in previous layers.

2.4.1 Multilayer feedforward network

Feedforward neural networks are built with one or more layers of fully connected (dense) nodes which connect to the following layers of nodes. Figure 7 shows an example. There are no recurrent connections to the previous layers. For a single-layer feedforward network, there is only the output layer of nodes which perform all the computations. It is capable of learning only linearly separable patterns. In the case of multilayer feedforward networks, the layers between the input and output layers are called ″hidden layers″. Its having multiple layers enables the network to learn also nonlinear patterns, such as a simple XOR function (Haykin 1999). The nodes are organized into groups of units called layers and they are usually arranged in a chain structure, where every layer is a function of the preceding layer (Goodfellow et al. 2016).

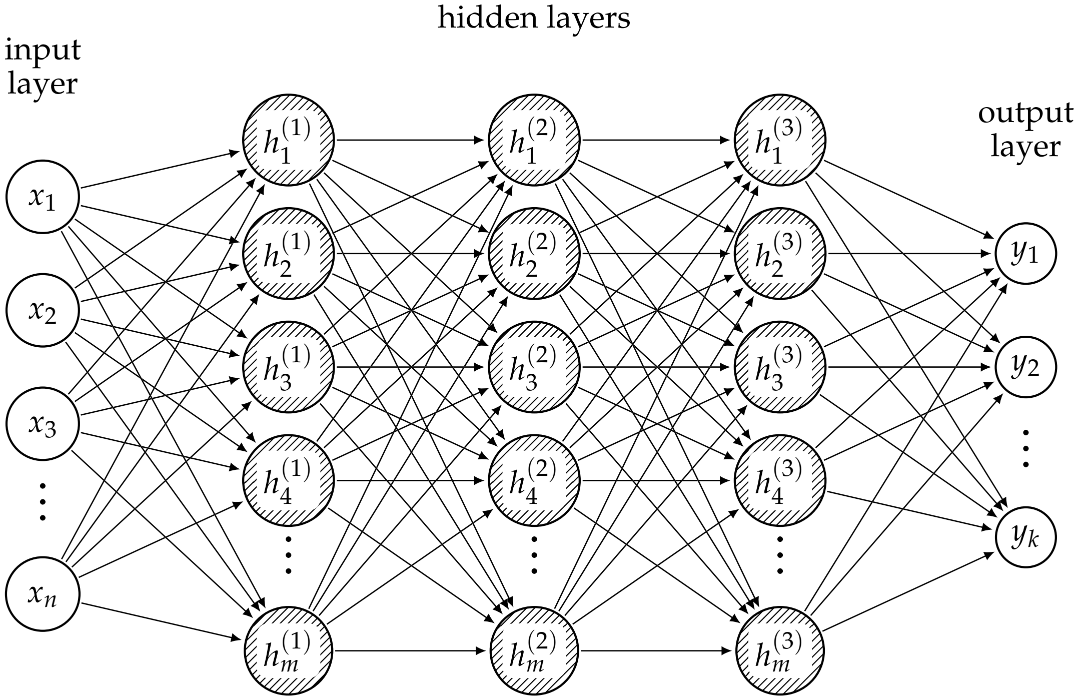


FIGURE 7 An example of a dense feedforward neural network with an input layer of n nodes, three hidden layers with m nodes each and an output layer with k output nodes. Inputs are denoted with x, outputs with y and hidden layer activations with $h^{(l)}$.

The first hidden layer activation is given by (Goodfellow et al. 2016):

$$h^{(1)}(x) = g^{(1)}(x) = g^{(1)}\left(\sum_{i=1}^{n} w_{1,i} \cdot x_i + b_1\right) \tag{2}$$

where x is the input vector and $w_{1,i}$ are the weights on the first layer. Subsequent hidden layer l is defined as a function of the preceding layers:

$$h^{(l)}(z) = g^{(l)}(z) = g^{(l)}\left(\sum_{i=1}^{n} w_{l,i} \cdot h^{(l-1)}(z_i) + b_l\right) \tag{3}$$

The previous layer's outputs z are given as inputs to the next layer. Figure 8 illustrates an activation of one neuron in a hidden layer.

For the network to be able to do inference on data, the weights of the nodes in the network have to be adjusted. Training data with desired target responses are fed to the network to adjust the weights of the nodes.

The algorithm for updating the network weight is the "error back-propagation algorithm". It is a method for computing the gradients of a network (Goodfellow et al. 2016). The actual learning is performed by an optimization algorithm, such as stochastic gradient descent. The back-propagation algorithm passes the different layers of the network twice: a forward pass and a backward pass. (Haykin 1999; Goodfellow et al. 2016)

In the forward pass, an input vector is applied to the network's source, and

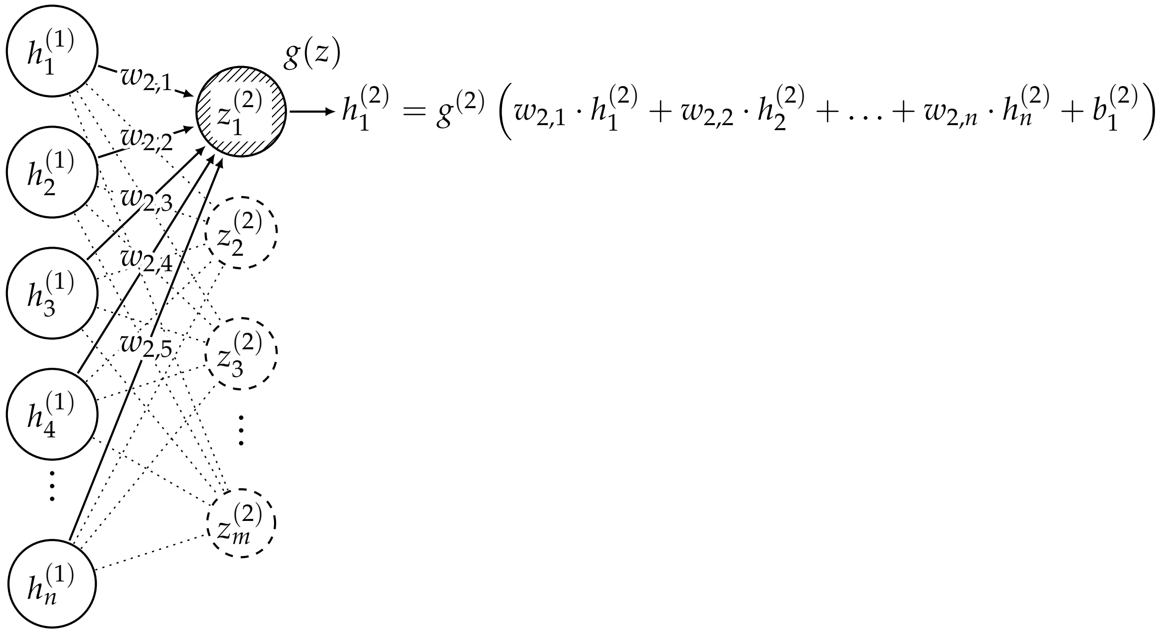


FIGURE 8 The output h of one neuron in a dense feedforward network layer. The activation function g takes the sum over the outputs weighted with the learnt coefficients w of the previous layer's neurons.

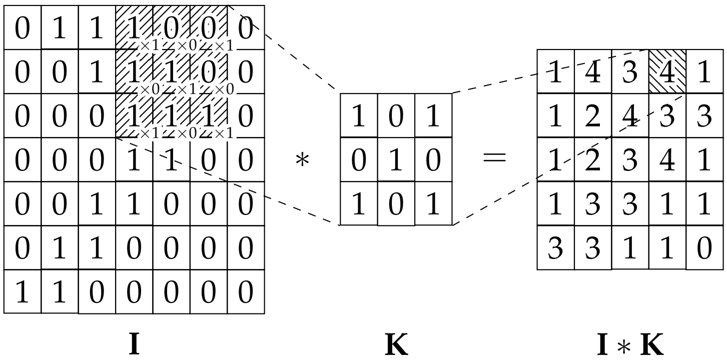
its effect propagates through the network until it reaches the output. The weights are fixed at this point.

The backward pass then adjusts the weights according to the errors. The error signals are calculated by calculating the loss function value from the actual response and the expected value. The error is propagated backwards through the network, updating the node weights. The weights are adjusted according to the partial derivatives to the direction that reduces the errors. (Haykin 1999; Goodfellow et al. 2016)

2.4.2 Convolutional neural network

A CNN can be described as a neural network that has at least one convolution operation. Generally, convolution is an integral that expresses the amount of overlap of a function x over another function w as it is shifted. Discrete convolution is denoted as:

$$s(t) = (x * w)(t) = \sum_{a=-\infty}^{\infty} x(a) \cdot w(t-a) \tag{4}$$



$$\begin{bmatrix}0&1&1&1&0&0&0\\0&0&1&1&1&0&0\\0&0&0&1&1&1&0\\0&0&0&1&1&0&0\\0&0&1&1&0&0&0\\0&1&1&0&0&0&0\\1&1&0&0&0&0&0\end{bmatrix} * \begin{bmatrix}1&0&1\\0&1&0\\1&0&1\end{bmatrix} = \begin{bmatrix}1&4&3&4&1\\1&2&4&3&3\\1&2&3&4&1\\1&3&3&1&1\\3&3&1&1&0\end{bmatrix}$$

FIGURE 9 2D convolution operation over image **I** using a 3×3 convolution kernel **K**.

where the first argument x is often referred to as the input and the second w as the kernel. The result of convolution is sometimes referred as the feature map. (Goodfellow et al. 2016)

Convolution is extendable to larger dimensions. Convolution has many applications in 2D image processing, such as edge detection, image blurring and interpolation. Processing a hyperspectral data cube with spatial and spectral dimensions is an example of using 3D convolutions. The challenge becomes selecting and defining a proper kernel for the use case.

Discrete convolution S in two dimensions is defined as (Goodfellow et al. 2016):

$$S(i,j) = (\mathbf{I} * \mathbf{K})(i,j) = \sum_{m}\sum_{n} \mathbf{I}(m,n) \cdot \mathbf{K}(i-m, j-n) \tag{5}$$

where **I** is the input and **K** is the convolutional kernel. Figure 9 illustrates a 2D convolution over an input. The convolutional kernel is spatially slided over the input and a feature map is computed. In the example, the output is smaller data array than the original, because convolution is executed only in the inner region of the input. Retaining the input dimension requires the edges to be handled separately or extended with padding. Padding techniques include zero-padding and mirroring data at the edge, but they can cause side effects and artifacts to the output. (Goodfellow et al. 2016)

A CNN differs from a dense feedforward neural network by how nodes interact with each other. In a dense network, every node is connected to all nodes in the previous layer. Convolutional nodes have sparse connectivity, which means that an output of a neuron is connected to only a part of the previous layer's nodes. Figure 10 depicts an example CNN where this kind of connectivity is visible. An example input image can have thousands or millions of pixels, but with sparse connectivity, it is possible to detect meaningful features with just tens or hundreds of pixels. This makes the computation more efficient computationally and in terms of memory requirements. (Goodfellow et al. 2016)

In a deep convolutional network, nodes in the deeper layers can indirectly affect with a large portion of the input. This means that the network output can take into account larger regions of the input. This receptive field can be further expanded with techniques, such as strided convolution and maxpooling func-

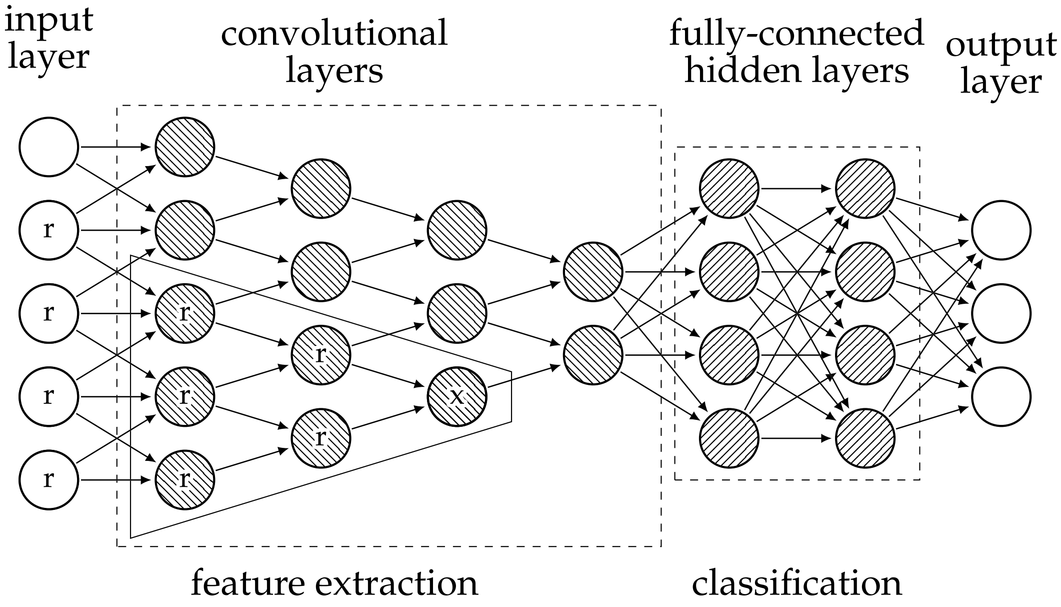


FIGURE 10 An example convolutional neural network with dense hidden layers and output for, e.g., classification. The neurons belonging to the receptive field of the convolutional neuron "x" are denoted with letter "r".

tions (Goodfellow et al. 2016). Figure 10 shows an example of a receptive field and the input nodes that are affected.

The kernel parameters are shared with the convolutional layer nodes, which decreases the model memory requirements drastically. Only one set of weights are learnt for all locations on a convolutional layer, in contrast to learning a separate set for every node in the case of a dense layer. (Goodfellow et al. 2016).

Stacked convolution operations construct a hierarchy of layers that detect features of increasingly abstract levels. In the first layers of the network, the learnt kernels are able to detect low-level features, such as edge patterns in an input image. Successive layers detect higher-level abstractions. As the same edges appear everywhere in the image, sharing weights across the entire image is practical. Pooling operations help make the representation approximately invariant to small translations of the input. This means the location of a feature in the image is irrelevant; rather only the existence of the feature is. This is a wanted property in, for example, image classification applications. (Goodfellow et al. 2016)

A common architecture for a CNN classifier is composed of two networks. The first part is a stack of feature extractors that employ convolutional layers, and the second part is a stack of fully-connected layers, which infers the final classification from given features, as depicted in Figure 10.

2.4.3 Image segmentation

Image segmentation can be defined as a classification problem in per pixel basis. Thus, it is regarded as a much harder problem than image classification, which assigns a single label for the entire image. Segmentation is further divided into semantic segmentation or instance segmentation. Semantic segmentation consid-

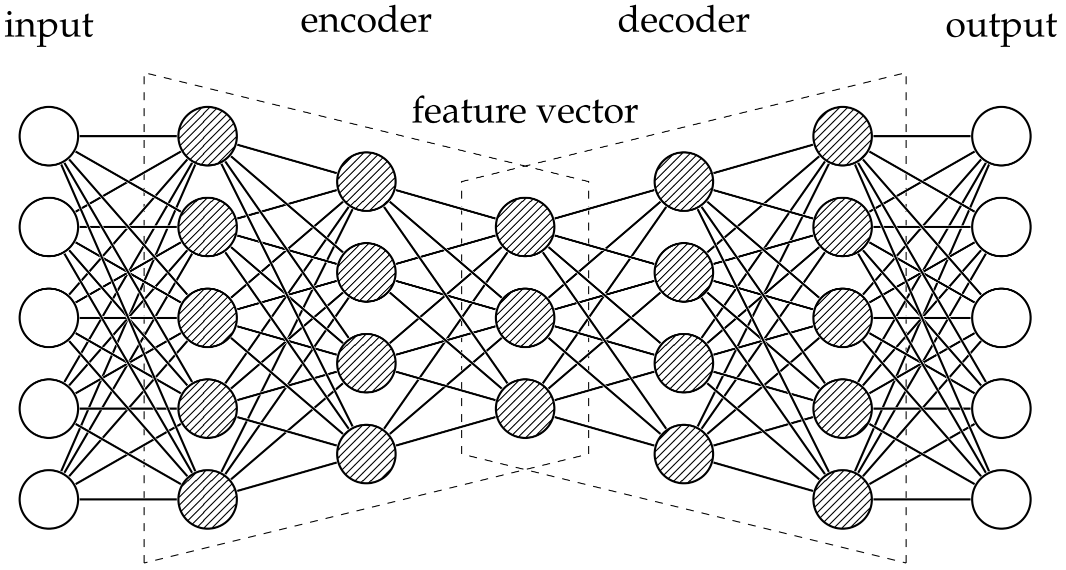


FIGURE 11 An example of dense encoder-decoder architecture.

ers a set of object categories, such as human, car and sky, for every pixel. Instance segmentation extends the scope to partitioning instances of each category, such as overlapping individual cars in the image. (Minaee et al. 2022)

The chosen segmentation method is subordinate to the used imaging modality and the application. Inferring a 2D segmentation from just 2D RGB data is achievable with enough well labeled data, but it is prone to ambiguities and error sources ranging from a lack lighting to the complexity of the captured scene. 2D segmentation accuracy can be increased by using 2.5D RGB-D images that include the depth as the fourth channel. Correspondingly, 3D segmentation refers to doing point-wise classification on volumetric data, such as point clouds or meshes. Segmentation have use cases in, e.g., robotics and 3D scene analysis. (Minaee et al. 2022)

The encoder-decoder deep learning architecture is often used for learning a more concise representation of a signal; it learns to map data from input to output by a two-stage network, as illustrated in Figure 11. Input data are compressed by the encoder to a latent-space representation, and the decoder attempts to predict the output from it. A latent-space representation means a feature vector smaller than the original data, and it holds the semantic information of the input (Minaee et al. 2022). This is similar to an autoencoder network that is trained to reproduce the input data to its output (Goodfellow et al. 2016; Minaee et al. 2022).

Many segmentation architectures have been developed on the encoder-decoder model, of which U-Net is one of the most well-known examples. It was originally developed for medical image segmentation, but it was later adopted in other fields. The encoder downsamples the path by extracting features with 3×3 convolutions and captures the context. The decoder expands the path with deconvolutions for precise localization. Skip connections between encoder and decoder are used to retain precise pattern information. U-Net is able to learn from a relatively small data set of images. U-Net and other encoder-decoder models do pixel-wise inference on whole images at a time. (Minaee et al. 2022)

2.4.4 CNN design for hyperspectral data

Designing a CNN for high-dimensional hyperspectral data exhibits some additional problems compared to designing for RGB data. A common problem is the size of the hyperspectral data cubes, which leads to increased memory requirements. Existing state-of-art RGB image segmentation methods take into account the larger context of the image. Also, the limited publicly available hyperspectral data sets make transfer learning unfeasible. Therefore, spectral neural networks have resorted to methods that consider dimension reduction and use only individual spectra or small patches of the data cube at a time (Wambugu et al. 2021; Paoletti et al. 2019). A classical way of reducing dimensionality is principle component analysis (Signoroni et al. 2019).

There exist methods for alleviating the over-learning and under-learning of hyperspectral models due to scarceness of training data. They include data augmentation for modifying the existing training data, synthetic data generation using, e.g., generative adversial networks, transfer learning and resorting to unsupervised or semi-supervised methods, where the model is first trained with unlabeled data and afterwards with a smaller labeled data set. (Wambugu et al. 2021)

Another challenge is intraclass variability; reflectance captured by a camera sensor includes artefacts and noise from the environment, such as changes in illumination and variation in athmospheric conditions. The spectrum may include significant redundancy in adjacent spectral bands which weaken the efficiency of the analysis algorithm. The imaged spectrum is usually mixed and can include spectra from many compounds, which introduce classification ambiguities that have to be handled by the algorithm. Interclass similarity is present especially in the border regions of different segmented classes. (Paoletti et al. 2019)

Older classification methods consider only the spectrum. For a CNN, this would mean using only 1D convolution operations. An extension to a per pixel approach would be to consider the neighborhood of each pixel. The sliding window deep neural network method applies a 2D kernel over of the input image (Paoletti et al. 2019). The classification is done only in the context of the window. With hyperspectral data, the spectral axis can also be taken in to account, and a 3D kernel can be moved inside spatial-spectral dimensions, as demonstrated by Zhang et al. (2018).

The sliding method is less demanding in terms of memory requirements and can be applied over large images. On the other hand, because the inference is executed for every image pixel position sequentially, the execution time can be long, and nodes of the network have more a limited receptive field to the whole image, which does not take larger input context into account.

Autoencoder models have been used in obtaining lower dimensional representations of hyperspectral data (Paoletti et al. 2019). Input and output carry the same data, and the constricted shape of the network forces the model to learn latent representation of the hyperspectral data, which can respectively be used by the analysis algorithm.

The fully convolutional spatial-spectral neural network architecture introduced by Long et al. (2015) segments a full hyperspectral data cube as a single input. The network consists of two stages: the first stage extracts features, and the second stage performs the mapping from the extracted features to a pixel-wise class map.

Still, due to the high dimensionality of the data, processing large data cubes with fully CNNs can be unfeasible as-is. One solution is splitting the input image, processing it in patches and combine the separate results to a final output.

Compared to a dense fully connected neural network, convolutional layers exhibit some advantages. The local connectivity enables learning spatial correlations in adjacent pixels and introduces approximate invariance to feature locations. Sparse connections save memory, and sharing parameters considers fewer weights to be fine-tuned during training, compared to a dense network architecture.

One recent state-of-art method for an encoder-decoder network for hyperspectral data cubes is the fast patch-free global learning framework. The framework introduced a designed sampling strategy and fully convolutional encoder-decoder network with skip connections to employ wider spatial information of the image. The system is designed to work with limited data, and it takes influences from image segmentation architectures, such as U-Net, employing similar skip connections. It aims to avoid redundant computations that the sliding window–based systems are prone to. (Zheng et al. 2020)

3 RESEARCH RESULTS AND RESEARCH CONTRIBUTION

3.1 PI: Convolutional neural networks in skin cancer detection using spatial and spectral domain

Results

In this work, different CNN architectures were experimented on to detect and classify different types of skin lesions for cancer diagnosis. The research was conducted with a small dataset (N=61) of hyperspectral images consisting of a set of several lesions. The neural networks were able to segment and classify regions from lesions. We experimented on 1D, 2D and 3D CNNs as well as combinations of 1D+3D and 1D+2D+3D architectures. The results showed promise, but the ground truth data were labelled based on the most dangerous lesion class in each image, which convoluted the result interpretation. A more general and reliable model is needed with more systematically gathered, diverse data set.

Author contribution

In this paper, the author contributed on the research with the original idea of using a sliding window type of architecture. The author experimented on different 2D and 3D CNNs for segmentation using the Python programming language and Keras software framework. The results showed promise, and the final design and experiments were conducted by Leevi Annala and Ilkka Pölönen.

Discussion

The proposed CNN architecture was fairly simple. The sliding-window architecture used is an inefficient technique, as it has to go through every pixel in the whole frame. Another weakness of the method is that the pixel-wise classifications do not take into account the larger spatial context of the image. The strengths of this method are that the training data can be generated easily from a small number of images and the GPU memory requirements are not high compared to a full-frame encoder-decoder network. However, full-frame encoder-decoder networks, like U-Net, have since been used in dermatologic diagnosis with hyperspectral data on mobile devices as demonstrated by La Salvia et al. (2022) with small image resolution. They can take into account a larger context in the image. These kinds of full-frame architectures were used in the authors's later research cases; in Paper PIII, U-Net was used for a similar medical segmentation problem and, in Paper PIV, a radiance spectra was approximated using frames of raw hyperspectral data.

This research answered research question Q1a on how to define a deep learning architecture for skin cancer classification. The results were used to design the network presented in Paper PIV. The paper also envisioned and paved the way for using surface normal data as extra features for the classification algorithm. A further study was made by Raita-Hakola et al. (2022) that presented how changing the illumination source can be used to capture photogrammetric stereo images and better the classification score, using a neural network similar to the one in this paper.

3.2 PII: Tree species identification using 3D spectral data and 3D convolutional neural network

Results

In this paper, we used a 3D CNN for identifying three Finnish tree species by using hyperspectral data and a digital surface model produced by dense image matching. The classification accuracies for each tree species were 96.2 % (Pine), 86.6 % (Spruce) and 98.2 % (Birch). The results show that a fairly simple 3D CNN could achieve good results in classifying hyperspectral data without manual feature extraction and selection.

Author contribution

In this article, the author influenced the creation of the used neural network, which was based on the work done in Paper PI, and contributed the idea of using saliency maps in the result verification. The author proofread the original draft and provided comments to improve the quality of the manuscript.

Discussion

This research answers research questions Q1b, on how to define a deep learning architecture for tree identification, and (partially) Q3 on how to fuse hyperspectral data with depth information. In the study, the point cloud was formed from a series of hyperspectral images taken with a flying UAV platform. The produced orthophotographic mosaic was used to extract regions around the tree canopies, and the depth information was concatenated as an additional layer to the hyperspectral data cube.

Generally, in a 2D CNN, each convolutional layer has a 2D convolution kernel for each input channel. In the case of a regular RGB image, there are three channels. Correspondingly, for a 3D CNN, there is a 3D convolution kernel for each channel. In this case, the channel count is one, as the kernel is slided through the grid space coordinates inside the spatial and spectral dimensions. The first 3D convolutional layer uses a 3D kernel shaped as (3, 3, 1), which means that the same kernel is used for all the spectral bands and the depth map. The kernel shape makes sure that only spatial patterns are taken into account in spite of the included imaging modalities. Consequent layers had (3, 3, 3) kernels that were applied on the feature maps from the previous layers. Surprisingly, the mixed multimodal nature of the input data did not seem to hinder the tree species inference, and the simple data concatenation approach seemed to work quite well.

Another approach could have been turning the trees into voxel grids with spectrum information in each index. In terms of memory usage, it could have been wasteful, but on the other hand the 3D CNN could possibly take the 3D shape of the tree better into account. Also, using two inputs, one for the HSI data cube and one for the depth map, could have been considered.

Saliency maps were used in the result verification, as neural networks are regarded as "black box" machine learning models. That means it is not well known what features of the input are actually emphasized in the inference. By calculating gradients over layers from output to input, it is possible to highlight the regions of the input image that contribute to the classification result. It could be seen from the top-down view that the tree canopies were highlighted. In the spectral dimension, the wavelengths contributed each species differently. These results indicate that the network took into account real tree features and not, for example, ground type.

3.3 PIII: Multilabel segmentation of cancer cell culture on vascular structures with deep neural networks

Results

The goal of the study was to find methods to automatize personalized drug efficacy assessment without actual test subjects. This work concerned segmentation of cells and structures from microscopy images of cell cultures. The major challenge of the research was the limited data set (N=36) and mutually non-exclusive classes. The same pixel could belong to multiple classes, as the cells and structures were both semitransparent and stacked on top of each other. We used multiple specialized U-Net encoder-decoder neural networks. The networks could distinguish between different structures, such as cells and spheroids. The results suggest that more diverse training data are needed for future research.

Author contribution

The author was responsible for all technical decisions regarding the computation aspects, such as modifying the used U-Net encoder-decoder convolutional neural network, preprocessing the data, defining and running the performance tests and interpreting the results. The author wrote the sections of the article concerning the computation.

Discussion

The author used U-Net neural network in this research, as it is known to learn from a small number of images to partition different semantic structures. Labeling this type of data was ambiguous, because it was not clear how, for example, the partially visible vascular structures would align under semitransparent cell mass. Using a sliding window would not have been an option in this case, as it was in Paper PI. We assumed that small RGB image patches do not include enough information for inferring their labels, in contrast to high-dimensional hyperspectral cube patches. We created a separate neural network for each label because each pixel could have multiple overlapping classes and the lack of data. As a result of this design choice, we did not force the network to learn what classes could occur together by having multiple multiclass activations in the output during the training. The author experimented with different activation functions, but they produced poor results.

This paper answered the research question Q2 on how well a CNN works in multilabel cell culture segmentation. Using this type of neural network architecture was a step towards Paper PIV where the neural network takes hyperspectral instead of RGB data.

3.4 PIV: Method for radiance approximation of hyperspectral data using deep neural network

Results

The author created a neural network that was able to approximate the radiance from the raw data produced by an FPI HSI camera. The proposed CNN delivered promising results with a limited data set (N=62) of hyperspectral images of patients with and without diagnosed melanoma cancer. The author defined a loss function that would take into account the shape and intensity of spectra. The method suffered from image artifacts in per band intensity fluctuations, but it was also noticed to reduce noise with noisy training data. This kind of network and loss function could potentially be used in other types of spectrum-generating neural networks.

Author contribution

The author was responsible for all technical decisions regarding the work, such as creating the CNN, preprocessing the data, defining and running the performance tests, interpreting the results and writing the manuscript.

Discussion

The neural network architectures in previous papers considered classification or segmentation problems where images or hyperspectral cubes were assigned labels on either a pixel, window or whole-image basis. The developed CNN takes a raw FPI HSI data cube and outputs a spectral radiance data cube. The predicted spectral radiance is defined as an array of continuous positive values, which affected the architecture and loss function design. This paper answers the research question Q1c on how to define a deep learning architecture for hyperspectral radiance approximation.

The created convolutional encoder-decoder network employs 3D convolutions operations in spatial dimensions to learn the bayer matrix interpolation function and spectral-wise convolutions for calculating the radiance. Usually, mean squared error (MSE) is used for regression networks. The problem with the MSE loss function was the unbalanced data, with the majority of the spectra representing healthy skin. The unbalance led the network to predict the same average spectrum for all pixels of the given input hyperspectral data cube. To retain the shape of spectra of the hyperspectral data cube, an additional cosine term was used in the loss function. This kind of architecture could be used in generative models to generate new model training data or other spectrum-modifying models, such as radiometric data correction.

3.5 PV: Reflectance Measurement Method Based on Sensor Fusion of Frame-based Hyperspectral Imager and Time-of-flight Depth Camera

Results

We demonstrated a method for fusing data from a Fabry-Perot interferometer hyperspectral camera and a Kinect V2 ToF depth sensing camera by using a set of calibration images. We created an experimental application utilizing the depth–augmented hyperspectral data to measure emission angle dependent reflectance from a multi-view inferred point cloud. The method could successfully combine 3D point clouds with hyperspectral data from different viewpoints. We estimated

3D surface normals of the target colorchecker board and calculated the emission angles. The results suggest that changing emission angle has a very small effect on surface reflectance intensity and spectrum shapes, which was expected with the used colorchecker. The work provides technical support for designing and implementing a system for hyperspectral 3D point cloud creation and analysis.

Author contribution

The author was responsible for the research as a whole; the original idea, writing the software, running the experiments, analyzing the data and writing most of the manuscript. Leevi Lind and Anna-Maria Raita-Hakola contributed in building the experimental setup and choosing the right optics as well as writing the optics and reflectance chapters of the manuscript. Leevi Lind wrote the radiance theory chapters and Sampsa Kiiskinen helped with the analysis methods. Ilkka Pölönen contributed in project administration and funding.

Discussion

The project started with an idea on how to combine these two cameras and later evolved to include analysis on how this kind of data could be applied. The original idea was to utilize the hyperspectral data to remove specular reflections, which occur on glossy surfaces, by using neural networks.

The author generated a data set from hyperspectral measurements that matched spectra from different viewing angles, designed a custom conditional 1D CNN and trained it in a supervised manner. The input of the system was the spectrum of a point on the colorchecker board with both the source and target emission angles. The output was the targeted spectrum with the corresponding emission angle. Using two halogen light sources positioned apart from each other should create a more significant specular component to the reflectance of the spectra, if the surface is glossy. The author tested another CNN model that was based on a conditional variational autoencoder design (Sohn et al. 2015). The architectures of the planned neural networks took influences from Paper PII and PIV.

After many experiments and attempts to remove the reflections using custom neural networks with the data, we noticed that the reflectance of the used colorchecker did not vary significantly angle-wise. In retrospect, the author should have used a differently shaped color reference object with more reflective optical properties and a round white reference object for improved angle-wise reflectance calculation.

The sensor fusion of HSI and the depth cameras was carried out using a calibration pattern and determining the intrinsic and extrinsic parameters. The calibration of the two cameras was the most time-consuming part of the work. Small misalignments and different optics of the cameras caused large errors in the estimated extrinsic parameters, and getting proper measurements required multiple attempts. Matching optics and robust attachment solved the problems.

This paper answers research question Q3 on how to fuse hyperspectral data with depth information in the context of ToF depth and frame-based HSI cameras.

4 CONCLUSIONS

Designing a neural network for processing hyperspectral data introduces some unique challenges compared to RGB images. Scarcely available training data, high dimensionality and size of individual data cubes remain problems. Camera hardware becoming more common and developments in neural network training will eventually diminish some of the current problems. In many applications, the speed of capture and processing have persisted as significant challenges.

This dissertation aims to answer questions on how to define deep learning architectures in the contexts of hyperspectral skin cancer classification, tree species identification, radiance approximation and medical cell culture RGB image segmentation. In Paper PIV, a deep learning neural network model was proposed to reduce the latency between data capture and ready-use radiance measurement. Similar kinds of takes on efficiency aspects of data processing were approached in Papers PI, PII and PIII, where data ambiguity, large data volumes and the supply of expert manual labeling work limit automatic utilization of data in large scales. The presented neural network applications displayed different levels of success, as the results showed promise in their narrow application spaces but lacked generalization due to small data sets. A fairly simple CNN can produce promising results. In Paper PIII, multilabel cell culture segmentation for RGB images introduced a new problem where the same pixel could include multiple classes simultaneously.

The CNN architecture has to be designed according to the application domain. In segmentation tasks, a large data cube has to be processed in smaller parts or per-pixel. A 2D CNN approach usually outperforms pixel-wise classifier as it takes the spatial domain better into account. 3D CNNs take the spectral domain into consideration, but tend to be slower and require more memory. In Paper PIV, the loss function had significant effect on the spectral radiance prediction performance.

Furthermore, to receive better inference results, depth information has previously shown its ability to improve classification in segmentation and classification tasks. Fusing hyperspectral data with depth data depends on the types of imaging modalities used. To answer the question on how to add depth information to hyperspectral data, Papers PII and PV involved augmenting hyperspectral data with depth information. Further study is needed for the sensor fusion application presented in Paper PV. The experimentally analysed target object did not support the goal of using the setup to improve reflectance measurements in the context of removing specular reflections. However, the results showed promise, and the paper provides technical support for designing and implementing a system for hyperspectral 3D point cloud creation and analysis. The applications of this dissertation are highly multidisciplinary, and the results include references for defining deep learning architectures for hyperspectral data.

Future work would include refining the hyperspectral and ToF depth data fusion by using a better color and white reference. A model, such as a neural network, and a hyperspectral point cloud could be used for minimizing specular reflections on glossy surfaces.

YHTEENVETO (SUMMARY IN FINNISH)

Neuroverkon suunnitteleminen hyperspektridatalle on haasteellista verrattuna tavallisiin värikuviin. Koulutusaineiston puute, tiedon moniulotteisuus ja yksittäisten datakuutioiden suuret koot ovat olleet ongelmia pitkään. Monilla sovellusalueilla kuvantamisen ja tiedon käsittelyn hitaus ovat merkittäviä ongelmia. Kuvantamiseen käytettyjen kameralaitteistojen yleistyminen ja neuroverkkojen kehitys tulevat kuitenkin lieventämään näitä ongelmia tulevaisuudessa.

Tämän väitöskirjan tavoite on vastata kysymykseen, kuinka määritellä syväoppivia arkkitehtuureja hyperspektrikuville ihosyövän luokitteluun, puulajien tunnistamiseen, radianssin likiarvoiseen määrittämiseen sekä soluviljelmien rajaukseen lääketieteellisistä värikuvista. Paperissa PIV esitettiin syväoppiva neuroverkkomalli, jolla pienennetään viivettä kuvan ottamisen ja valmiin radianssimittauksen välillä. Vastaavanlaista tiedonkäsittelyn tehokkuuteen tähtäävää lähestymistapaa ehdotettiin myös Papereissa PI, PII ja PIII. Näissä sovelluksissa tiedon monitulkintaisuus, suuret tietomassat ja tarve käsin tehtävälle asiantuntijatyölle ovat rajoittaneet automaattista tiedon hyödyntämistä suuressa mittakaavassa. Esitetyt neuroverkot osoittivat vaihtelevia tuloksia rajatuilla sovellusalueillaan. Paperissa PIII esitetty menetelmä RGB-kuvien pikselikohtaiselle luokkiin jakamiselle toi esiin uuden ongelman: kuvan pikseli saattoi kuulua useaan eri luokkaan samanaikaisesti.

Konvoluutioneuroverkon arkkitehtuuri täytyy suunnitella sovelluksen mukaisesti. Pikselikohtaisessa luokittelussa suuret datakuutiot täytyy käsitellä pienissä osissa tai pikseli kerrallaan. Kaksiulotteinen konvoluutioverkko pärjää näissä tehtävissä yleensä paremmin kuin pikselikohtainen luokittelija, koska se huomioi pistettä ympäröivät spektrit paremmin. Kolmiulotteinen konvoluutioverkko ottaa spektridimension huomioon, mutta vaatii enemmän muistia ja on yleensä hitaampi. Paperissa PIV virhefunktiolla havaittiin olevan suuri merkitys spektriradianssin likiarvoiseen määrittämiseen.

Syvyystietoa on käytetty aiemmin parempien luokittelutulosten tuottamiseen kuva- ja pikselitasolla. Syvyys- ja hyperspektritiedon yhdistäminen riippuu käytetyistä kuvantamismenetelmistä. Paperit PII ja PV pyrkivät vastaamaan tutkimuskysymykseen, kuinka yhdistää syvyystietoa hyperspekstrikuviin. Tulokset olivat lupaavia, mutta jatkotutkimuksia tarvitaan Paperissa PV esitetylle sensorifuusiolle. Kokeessa ei pystytty poistamaan spekulaariheijastumia kohteesta otetusta kuvasta. Tutkimus tarjoaa kuitenkin ohjeita ja työkaluja vastaavanlaisen hyperspektripistepilven rakentamiseen ja analysointiin. Väitöskirjassa esitetyt sovellukset ovat luonteeltaan poikkitieteellisiä ja tarjoavatkin esimerkkejä syväoppivien arkkitehtuurien suunnitteluun hyperspektridatalle.

Esimerkki jatkotutkimuksesta olisi hyperspektri- ja syvyysdatan yhdistäminen käyttäen parempia väri- ja valkoreferenssejä. Neuroverkkoa ja spektrin sisältävää pistepilveä voitaisiin mahdollisesti hyödyntää spekulaariheijastusten poistamiseen kiiltäviltä pinnoilta.

REFERENCES

Annala, L. & Pölönen, I. 2022. Kubelka–Munk Model and Stochastic Model Comparison in Skin Physical Parameter Retrieval. doi:10.1007/978-3-030-70787-3_10. ⟨URL:https://doi.org/10.1007/978-3-030-70787-3_10⟩.

Audebert, N., Le Saux, B. & Lefevre, S. 2019. Deep learning for classification of hyperspectral data: A comparative review. IEEE Geoscience and Remote Sensing Magazine 7 (2), 159-173. doi:10.1109/MGRS.2019.2912563.

Bayarri, V., Sebastián, M. A. & Ripoll, S. 2019. Hyperspectral imaging techniques for the study, conservation and management of rock art. Applied Sciences 9 (23). doi:10.3390/app9235011. ⟨URL:https://www.mdpi.com/2076-3417/9/23/5011⟩.

Bioucas-Dias, J. M., Plaza, A., Camps-Valls, G., Scheunders, P., Nasrabadi, N. & Chanussot, J. 2013. Hyperspectral remote sensing data analysis and future challenges. IEEE Geoscience and Remote Sensing Magazine 1 (2), 6-36. doi:10.1109/MGRS.2013.2244672.

Bodkin, A., Sheinis, A., Norton, A., Daly, J., Beaven, S. & Weinheimer, J. 2009. Snapshot hyperspectral imaging: the hyperpixel array camera. In S. S. Shen & P. E. Lewis (Eds.) Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XV, Vol. 7334. SPIE, 73340H. doi:10.1117/12.818929. ⟨URL:https://doi.org/10.1117/12.818929⟩.

Chang, C.-I. 2007. Hyperspectral Data Exploitation: Theory and Applications. USA: Wiley-Interscience.

Eskelinen, M. A. 2019. Computational methods for hyperspectral imaging using Fabry–Perot interfer-ometers and colour cameras. University of Jyväskylä. Ph. D. Thesis.

Gokturk, S., Yalcin, H. & Bamji, C. 2004. A time-of-flight depth sensor - system description, issues and solutions. In 2004 Conference on Computer Vision and Pattern Recognition Workshop, 35-35. doi:10.1109/CVPR.2004.291.

Goodfellow, I., Bengio, Y. & Courville, A. 2016. Deep Learning. MIT Press. ⟨URL:http://www.deeplearningbook.org⟩.

Grusche, S. 2014. Basic slit spectroscope reveals three-dimensional scenes through diagonal slices of hyperspectral cubes. Applied Optics 53 (20), 4594–4603. doi:10.1364/AO.53.004594. ⟨URL:https://opg.optica.org/ao/abstract.cfm?URI=ao-53-20-4594⟩.

Hansard, M., Lee, S., Choi, O. & Horaud, R. P. 2012. Time-of-flight cameras: principles, methods and applications. Springer Science & Business Media.

Hapke, B. 2012. Theory of Reflectance and Emittance Spectroscopy (2nd edition). Cambridge University Press. doi:10.1017/CBO9781139025683. ⟨URL:https://www.cambridge.org/core/books/theory-of-reflectance-and-emittance-spectroscopy/C266E1164D5E14DA18141F03D0E0EAB0⟩.

Hauser, J., Zheludev, V. A., Golub, M. A., Averbuch, A., Nathan, M., Inbar, O., Neittaanmäki, P. & Pölönen, I. 2017. Snapshot spectral and color imaging using a regular digital camera with a monochromatic image sensor. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XLII-3/W3, 51–58. doi:10.5194/isprs-archives-XLII-3-W3-51-2017. ⟨URL:https://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/XLII-3-W3/51/2017/⟩.

Haykin, S. 1999. Neural Networks: A Comprehensive Foundation. Prentice Hall.

Henke, M., Junker, A., Neumann, K., Altmann, T. & Gladilin, E. 2018. Automated alignment of multi-modal plant images using integrative phase correlation approach. Frontiers in Plant Science 9, 1519.

Iglhaut, J., Cabo, C., Puliti, S., Piermattei, L., O'Connor, J. & Rosette, J. 2019. Structure from motion photogrammetry in forestry: a review. Current Forestry Reports 5 (3), 155-168. doi:10.1007/s40725-019-00094-3. ⟨URL:https://doi.org/10.1007/s40725-019-00094-3⟩.

La Salvia, M., Torti, E., Leon, R., Fabelo, H., Ortega, S., Balea-Fernandez, F., Martinez-Vega, B., Castaño, I., Almeida, P., Carretero, G., Hernandez, J. A., Callico, G. M. & Leporati, F. 2022. Neural networks-based on-site dermatologic diagnosis through hyperspectral epidermal images. Sensors 22 (19). doi:10.3390/s22197139. ⟨URL:https://www.mdpi.com/1424-8220/22/19/7139⟩.

Lillesand, T., Kiefer, R. & Chipman, J. 2007. Remote Sensing and Image Interpretation (6th edition). John Wiley & Sons. doi:10.2307/634969.

Long, J., Shelhamer, E. & Darrell, T. 2015. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Lu, Y., Saeys, W., Kim, M., Peng, Y. & Lu, R. 2020. Hyperspectral imaging technology for quality and safety evaluation of horticultural products: A review and celebration of the past 20-year progress. Postharvest Biology and Technology 170, 111318. doi:https://doi.org/10.1016/j.postharvbio.2020.111318. ⟨URL:https://www.sciencedirect.com/science/article/pii/S0925521420308905⟩.

Minaee, S., Boykov, Y., Porikli, F., Plaza, A., Kehtarnavaz, N. & Terzopoulos, D. 2022. Image segmentation using deep learning: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence 44 (7), 3523-3542. doi:10.1109/TPAMI.2021.3059968.

Nascimento, J. & Dias, J. 2005. Vertex component analysis: a fast algorithm to unmix hyperspectral data. IEEE Transactions on Geoscience and Remote Sensing 43 (4), 898-910. doi:10.1109/TGRS.2005.844293.

Okamoto, T. & Yamaguchi, I. 1991. Simultaneous acquisition of spectral image information. Optics Letters 16 (16), 1277–1279. doi:10.1364/OL.16.001277. ⟨URL:https://opg.optica.org/ol/abstract.cfm?URI=ol-16-16-1277⟩.

Paoletti, M., Haut, J., Plaza, J. & Plaza, A. 2019. Deep learning classifiers for hyperspectral imaging: A review. ISPRS Journal of Photogrammetry and Remote Sensing 158, 279-317. doi:https://doi.org/10.1016/j.isprsjprs.2019.09.006. ⟨URL:https://www.sciencedirect.com/science/article/pii/S0924271619302187⟩.

Puupponen, H.-H. 2014. Unmixing methods in novel applications of spectral imaging. University of Jyväskylä. Ph. D. Thesis.

Raita-Hakola, A.-M., Annala, L., Lindholm, V., Trops, R., Näsilä, A., Saari, H., Ranki, A. & Pölönen, I. 2022. Fpi based hyperspectral imager for the complex surfaces calibration, illumination and applications. Sensors 22 (9). doi:10.3390/s22093420. ⟨URL:https://www.mdpi.com/1424-8220/22/9/3420⟩.

Rosnell, T. & Honkavaara, E. 2012. Point cloud generation from aerial image data acquired by a quadrocopter type micro unmanned aerial vehicle and a digital still camera. Sensors 12 (1), 453–480. doi:10.3390/s120100453. ⟨URL:https://www.mdpi.com/1424-8220/12/1/453⟩.

Saari, H., Aallos, V.-V., Akujärvi, A., Antila, T., Holmlund, C., Kantojärvi, U., Mäkynen, J. & Ollila, J. 2009. Novel miniaturized hyperspectral sensor for UAV and space applications. In R. Meynart, S. P. Neeck & H. Shimoda (Eds.) Sensors, Systems, and Next-Generation Satellites XIII, Vol. 7474. SPIE, 74741M. doi:10.1117/12.830284. ⟨URL:https://doi.org/10.1117/12.830284⟩.

Saari, H., Pölönen, I., Salo, H., Honkavaara, E., Hakala, T., Holmlund, C., Mäkynen, J., Mannila, R., Antila, T. & Akujärvi, A. 2013. Miniaturized hyperspectral imager calibration and uav flight campaigns. In Sensors, systems, and next-generation satellites xvii, Vol. 8889. SPIE, 448–459.

Sandak, J., Sandak, A., Legan, L., Retko, K., Kavčič, M., Kosel, J., Poohphajai, F., Diaz, R. H., Ponnuchamy, V., Sajinčič, N., Gordobil, O., Tavzes, Č. & Ropret, P. 2021. Nondestructive evaluation of heritage object coatings with four hyperspectral imaging systems. Coatings 11 (2). doi:10.3390/coatings11020244. ⟨URL:https://www.mdpi.com/2079-6412/11/2/244⟩.

Sell, J. & O'Connor, P. 2014. The xbox one system on a chip and kinect sensor. IEEE Micro 34 (2), 44-53. doi:10.1109/MM.2014.9.

Signoroni, A., Savardi, M., Baronio, A. & Benini, S. 2019. Deep learning meets hyperspectral image analysis: A multidisciplinary review. Journal of Imaging 5

(5). doi:10.3390/jimaging5050052. ⟨URL:https://www.mdpi.com/2313-433X/5/5/52⟩.

Sohn, K., Lee, H. & Yan, X. 2015. Learning structured output representation using deep conditional generative models. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama & R. Garnett (Eds.) Advances in Neural Information Processing Systems, Vol. 28. Curran Associates, Inc. ⟨URL:https://proceedings.neurips.cc/paper/2015/file/8d55a249e6baa5c06772297520da2051-Paper.pdf⟩.

Striova, J., Dal Fovo, A. & Fontana, R. 2020. Reflectance imaging spectroscopy in heritagescience. La Rivista del Nuovo Cimento 43 (10), 515-566. doi:10.1007/s40766-020-00011-6. ⟨URL:https://doi.org/10.1007/s40766-020-00011-6⟩.

Sun, G., Wang, X., Sun, Y., Ding, Y. & Lu, W. 2019. Measurement method based on multispectral three-dimensional imaging for the chlorophyll contents of greenhouse tomato plants. Sensors 19 (15). doi:10.3390/s19153345. ⟨URL:https://www.mdpi.com/1424-8220/19/15/3345⟩.

Szeliski, R. 2022. Computer vision: algorithms and applications. Springer Nature.

Toivonen, M. E., Rajani, C. & Klami, A. 2020. Snapshot hyperspectral imaging using wide dilation networks. Machine Vision and Applications 32 (1), 9. doi:10.1007/s00138-020-01136-8. ⟨URL:https://doi.org/10.1007/s00138-020-01136-8⟩.

Wambugu, N., Chen, Y., Xiao, Z., Tan, K., Wei, M., Liu, X. & Li, J. 2021. Hyperspectral image classification on insufficient-sample and feature learning using deep neural networks: A review. International Journal of Applied Earth Observation and Geoinformation 105, 102603. doi:https://doi.org/10.1016/j.jag.2021.102603. ⟨URL:https://www.sciencedirect.com/science/article/pii/S030324342100310X⟩.

Zhang, X., Sun, Y., Zhang, J., Wu, P. & Jiao, L. 2018. Hyperspectral unmixing via deep convolutional neural networks. IEEE Geoscience and Remote Sensing Letters 15 (11), 1755-1759. doi:10.1109/LGRS.2018.2857804.

Zheng, Z., Zhong, Y., Ma, A. & Zhang, L. 2020. Fpga: Fast patch-free global learning framework for fully end-to-end hyperspectral image classification. IEEE Transactions on Geoscience and Remote Sensing 58 (8), 5612-5626. doi:10.1109/TGRS.2020.2967821.

ORIGINAL PAPERS

PI

CONVOLUTIONAL NEURAL NETWORKS IN SKIN CANCER DETECTION USING SPATIAL AND SPECTRAL DOMAIN

by

Ilkka Pölönen, **Samuli Rahkonen**, Leevi Annala and Noora Neittaanmäki 2019

Proceedings of SPIE Volume 10851: Photonics in Dermatology and Plastic
Surgery, SPIE, The International Society for Optical Engineering

Convolutional neural networks in skin cancer detection using spatial and spectral domain

Ilkka Pölönen[a], Samuli Rahkonen[a], Leevi Annala[a], and Noora Neittaanmäki[b]

[a]Faculty of Information Technology, University of Jyväskylä, Mattilanniemi 2, Jyväskylä, Finland
[b]Departments of Pathology and Dermatology, Institutes of Biomedicine and Clinical Sciences, Sahlgrenska Academy, University of Gothenburg, Gothenburg, Sweden

ABSTRACT

Skin cancers are world wide deathly health problem, where significant life and cost savings could be achieved if detection of cancer can be done in early phase. Hypespectral imaging is prominent tool for non-invasive screening. In this study we compare how use of both spectral and spatial domain increase classification performance of convolutional neural networks. We compare five different neural network architectures for real patient data. Our models gain same or slightly better positive predictive value as clinicians. Towards more general and reliable model more data is needed and collection of training data should be systematic.

Keywords: Hyperspectral imaging, convolutional neural network, skin cancer, melanoma

1. INTRODUCTION

Skin cancers are constantly increasing problem world wide. Traditionally this has been concern of people whose skin is relatively lightly coloured and annual portion of sunlight is high. Because of increased traveling and ageing of the population, melanoma is increasing problem also in the Nordic countries. For example in Sweden,[1] 50 % of all the annual skin cancer related costs are caused by melanomas.

There is a need for tools, which are able to detect early stage skin cancers and delineate them properly from healthy tissue. With proper detection it is possible to reduce amount of re-surgeries, when part of the malignant tissue has been left to the patient in original tumor removal. This is highlighted by the fact that overall positive predictive value of clinical melanoma diagnosis is 33 %.[2] In non-specialised clinics this is even lower. For every melanoma removal there will be 9 to 30 non-melanoma lesions removed depending on how specialised clinic is.[3] Thus, early detection will lower the treatment costs and will ensure higher survival rate.

Hypersepctral imaging is method where hundreds narrow wavebands of light are imaged simultaneously. This method will provide almost continuous spectrum for each pixel of the image as figure 1 is showing. Hyperspectral imaging is non-invasive imaging modality, because it is using only visible and near infra-red illumination to capture images. Previously we have used it in delineation of tumor border and distinguish in-situ melanoma from malignant melanoma.[4,5]

If you look at closely two spectra in the figure 1 , it is quite easy to see that in clear cases melanoma and healthy skin have characteristic spectra. Unfortunately this is not so in all the cases. In figure 2 we have spectral distributions of malignant melanoma, lentigo-maligna, dysplastic nevus and benign nevus. We can see that these distributions are overlapping. This means that if the melanoma is hard to recognise in clinical study, it will be hard distinguish using just spectral information. Thus, it seems natural that we also utilize spatial domain in the classification task.

Further author information: (Send correspondence to Ilkka Pölönen)
Ilkka Pölönen: E-mail: ilkka.polonen@jyu.fi, Telephone: +358 400 248 140
Samuli Rahkonen: E-mail: samuli.rahkonen@jyu.fi
Leevi Annala: E-mail: leevi.a.annala@jyu.fi
Noora Neittaanmäki: E-mail: noora.neittaanmaki@fimnet.fi

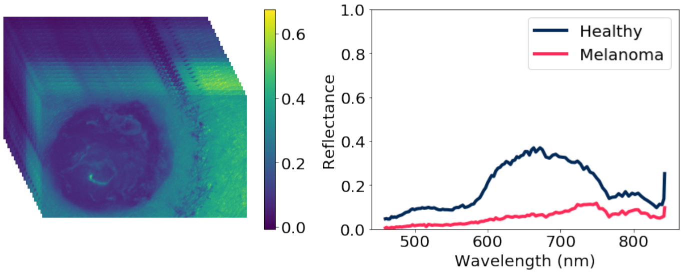


Figure 1. Hyperspectral image consist of intensity images, which are narrow wavebands from visible and near-infra red region of the electromagnetic spectrum. Each pixel in the image is spectrum through spectral domain.

Convolutional neural networks have shown great success in different kind of pattern recognition tasks.[6–8] They have also been recently used in classifying melanomas and other skin cancers from dermatoscope and regular color images.[9] In these cases, results are given for the whole images. Because of such binary classification we don't actually have an opportunity to determinate lesion's borders from analysis or what kind of other irregularities there are in the tumor.

There are multiple strategies to utilize convolutional neural networks. We are introducing efficient strategy, which contains utilization of both spectral and spatial domain. With hyperspectral data containing wavebands from visible to infra-red region, we are able to gather more information from each pixel than using regular imaging systems.[4,5] Using sliding window method over captured spectral image, we will have spectral and spatial domain for further analysis. In this study we describe some incremental steps, which are taking us closer to automatic skin cancer detection, identification and delineation.

2. MATERIAL AND METHODS

In this study we have small data set (n=61) of hyperspectral images covering narrow wavebands from 450 to 850 nm. Data set consist of several lesions, which were imaged and diagnosed by histopathology. Lesions consist of malignant melanomas, melanoma in-situs, dysplastic nevi and bening nevi. All patients have volunteered to participate in the study. The study protocol has followed the Declaration of Helsinki and it was approved by the local Ethics committee. Patient were recruited and imaged by the Department of Dermatology and Allergology of Helsinki University Hospital, Helsinki, Finland and by the Päijät-Häme Central Hospital, Lahti, Finland, between June 2016 and October 2017.

All hyperspectral images were collected with two identical hyperspectral imagers (Revenio Prototype 2016). Spectral separation of the imager is based on Fabry-Pérot interferometer (FPI). Use of FPI enables fast scanning in the spectral domain. The imager works on wavebands from 450 nm to 850 nm. The imager captures 120 wavebands within few seconds. Full width of each waveband's half maximum (FWHM) vary from 5 to 15 nm. Variation in FWMH comes as a function of wavelength. Another source of the variation comes from which multiple of FPI's is used. Imaging system contains a broadband halogen light source, which produces diffuse illumination to the imaged region of the interest (ROI). At the imager there is covering tube, which blocks illumination from other sources. Image acquisition is done with color cmos machine vision camera, which is integrated to the imager. The used machine vision camera is capable to take images in 1920 × 1200 pixel resolution. This corresponds approximately to 15 μm/pixel spatial resolution.

The spectral imager produces a raw data cube, which is calibrated to the radiance by following method of Saari et. al (2013).[10] There was some indeterminated fluctuation at the end of recorded spectra. Thus, twenty last wavebands were left outside from further analysis. For each data cube there was captured white reference target. This was used to convert imaged radiance to reflectance $R = I/I_0$, where I is imaged region of interest and I_0 is data cube from white reference. To improve quality of the data in spectral domain and reduce memory consumption in further processing, the data is downsampled. This was done by averaging nearest pixels of every fifth pixel. Also, only every second waveband was used in further analysis. By these operations data cubes size reduced to 384 × 240 × 50 pixels.

Training of the classifier needs labelled data. For each image there were annotated areas, which indicated either healthy skin, lesion or used marker. From each image's annotated areas 1000 data sets (or less if annotated area contained less than 1000 pixels) were selected for training purposes. These data sets contained annotated pixel and its 10 × 10 neighborhood. Figure 2 shows distribution of spectra of melanoma, lentigo maligna, dysplastic nevus and benign nevus. As we can see that there are overlapping in the distributions and some of the spectra has deviation. To reduce these effects and some problems from the vignetting and lightning irregularities, each imaged spectrum was subtracted by its average in spectral domain.

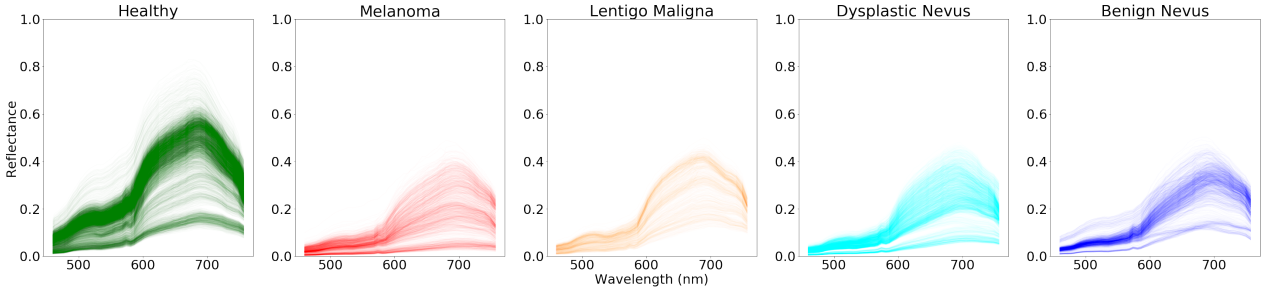


Figure 2. The distributions of spectra of melanoma, lentigo maligna, dysplastic nevus and benign nevus.

During recent years, deep neural networks have made new records in pattern recognition.[6] Our aim was to use both spatial and spectral domain simultaneously. Convolutional neural networks have been used to classify melanomas and other skin cancers from dermatoscope and regular color images.[9] Spectral data cube has three-dimensional nature, thus, standard 2D convolutional neural network might not be enough to utilize spectral data.

In deep learning and especially with convolutional neural networks classification task has two parts. In the feature learning we are calculating features using convolution operations with different weights. By tuning weights during back-propagation we will eventually achieve optimized feature space for our classification task. The actual classification model is just a deep regular multi-layer perceptron network. This structure is illustrated in the figure 3.

In this study we tested three different kind of feature learning structures - 1D, 2D and 3D convolutions. We will also have basically two different types of inputs. Single spectra and small window surrounding this spectra. As figure 3 shows, a 1D convolution input takes a single spectra. For 2D and 3D convolution input will be a subset's of spectral cube. Difference between 2D and 3D convolution is that 2D case doesn't operate over spectral domain while 3D does.

Used deep neural networks consist from two parts. First part executes feature learning by using the convolutional operator by the Conv layers. The Maxpooling layers reduces data's dimensionality. Machine learning part and actual classification is done with deep feed-forward neural network, which consist of six Dense layers. Convolutional and dense layers use rectified linear unit activation (ReLU) function. The single Dropout layer is added to avoid overfitting of the model. Last dense layer does final classification using Softmax activation function. Parameters of each layer are shown in the figure 3.

By variating the described architecture we tested five different kind of networks - 1D, 2D and 3D convolutional neural networks and two combinations where feature learning was executed by using 3D+1D convolutions and 3D+2D+1D convolutions.

The actual training data was sampled randomly from annotated points, so that there were 10000 data points from each class. An annotation was based on its histopathological results. Whole lesions were marked the same way. Annotation was done by a non-expert.

For annotated points, data augmentations was utilized so, that each training cube was mirrored and flipped horizontally and vertically. These operations fourfold the number of the inputs in the training phase. Training

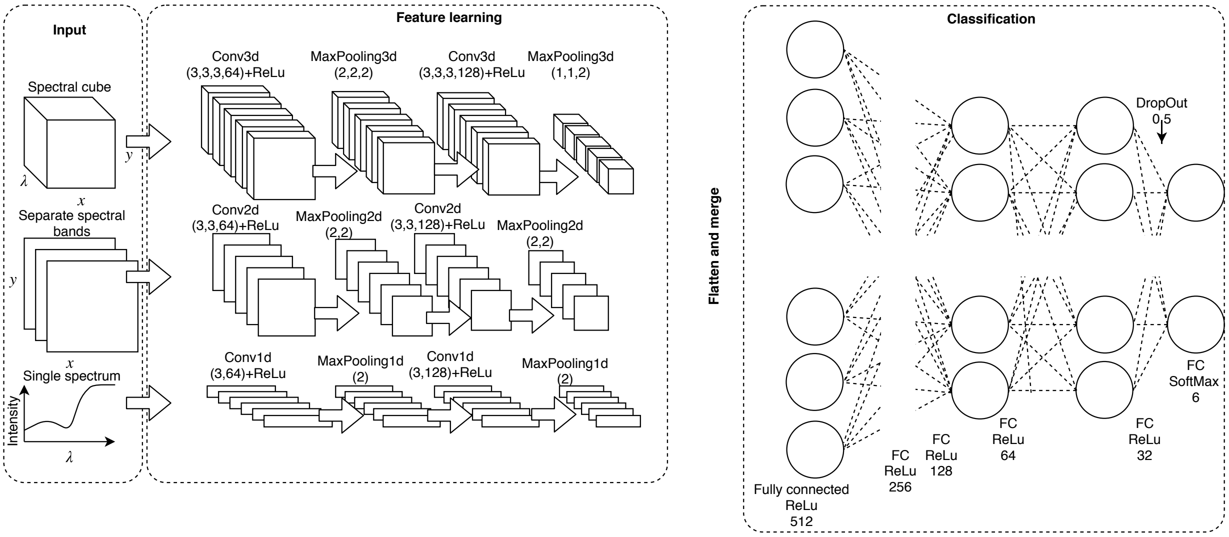


Figure 3. Schematic structure of used convolutional neural networks. Best results were gained using all inputs and all three different convolutional feature learning parts simultaneously.

set consisted of approximately 240 000 data points. For the optimization we used Adam, which is a first-order gradient-based optimization method of stochastic objective functions. The used hyperparameters for the optimization was the learning rate of 0.001, $\beta_1 = 0.9$ and $\beta_2 = 0.999$, while the learning rate decay over each update stayed at 0. The used cost function was categorical cross-entropy.

Our implementation used Keras with Tensorflow backend and Python 3.6 . All calculations were executed using IBM PowerAI platform, which includes two Nvidia Tesla V100-SXM2 16 GB GPU units.

There were only 61 imaged lesions (15 malignant melanoma, 6 lentigo maligna, 26 dysplastic nevus and 14 benign nevus). Thus, leave-one-out cross-validation was used. In this procedure classifier is trained 61 times for each image separately. This will guarantee that training set does not include data points from the image which is currently under classification.

3. RESULTS

Our ground truth consist of the results of histopathology. This meant that whole lesion was labelled based on most dangerous diagnosis. Because our approach gives us pixel wise information we ended situations where one lesion had several differently classified pixels. This actually might be quite realistic situation. Malignant lesions can have non-malignant parts. Thus, final classification for each lesion was made based on most dangerous pixel, which was found from lesion. If there was even a single pixel, which was classified as melanoma, whole lesion was classified to melanoma. In melanoma detection with this approach we will gain relatively high sensitivity, but low specificity which is seen in table 1 and in figure 8. And as opposite for benign nevus will have high specificity and low sensitivity.

Table 1. Sensitivity, specificity and positive predictive value of different classifiers for the melanoma classification

	CNN 1D	CNN 2D	CNN 3D	CNN 3D+1D	CNN 3D+2D+1D
Sensitivity	1	1	0.93	0.93	0.93
Specificity	0.15	0.12	0.14	0.14	0.21
Positive predictive value	0.34	0.35	0.32	0.32	0.34

When we are looking at sensitivities of different classifiers, we can see, that all 15 melanoma cases actually were classified correctly using only 1D and 2D convolution networks. 14 of 15 melanoma cases were classified correctly when 3D convolution was utilised. These metrics are actually misleading. If we look at actual classification results as shown in figures 4, 5, 6 and 7, we can see that actually classification results based on single spectra are often noisy. Figure 4 was confirmed to be dysplastic nevus in histopathology. All single source convolutional neural networks fail to classify it correctly. On lesion boundaries there is quite typical error, where trained model for some reason mis-classify lesion to melanoma. Here best result is achieved using multiple inputs and three different kind on CNN's.

In general level it seems that spatial features give more reliable looking results. When we combine those with spectral domain, results get better, because specificity increases. If we take closer look one false positive case in the figure 7 we can see that majority of the pixels in the lesion is actually classified correctly. This is promising result because with more training cases we might have chance to train better models.

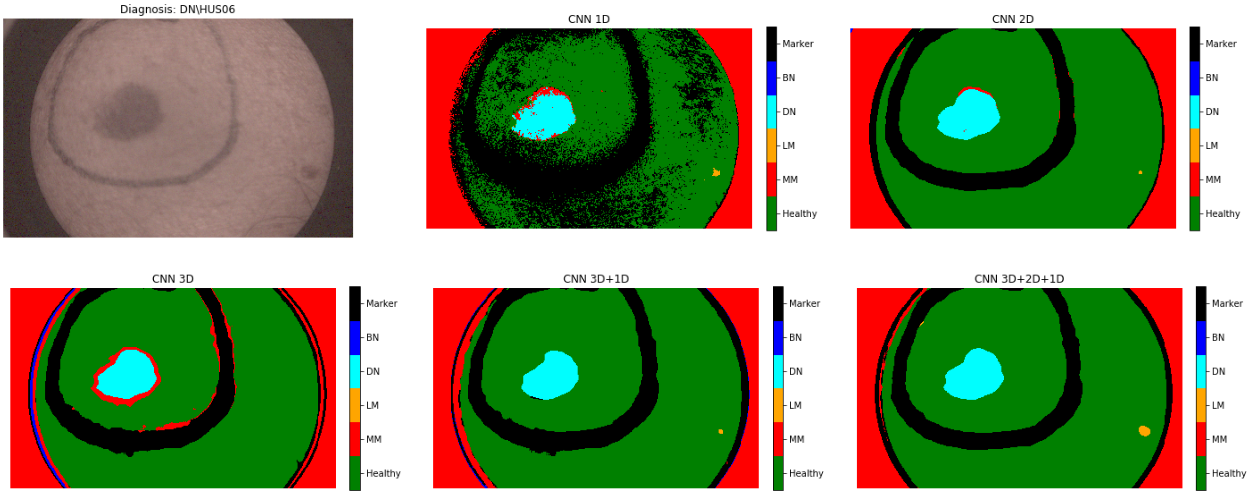


Figure 4. Classification results of five different classifiers for dysplastic nevus.

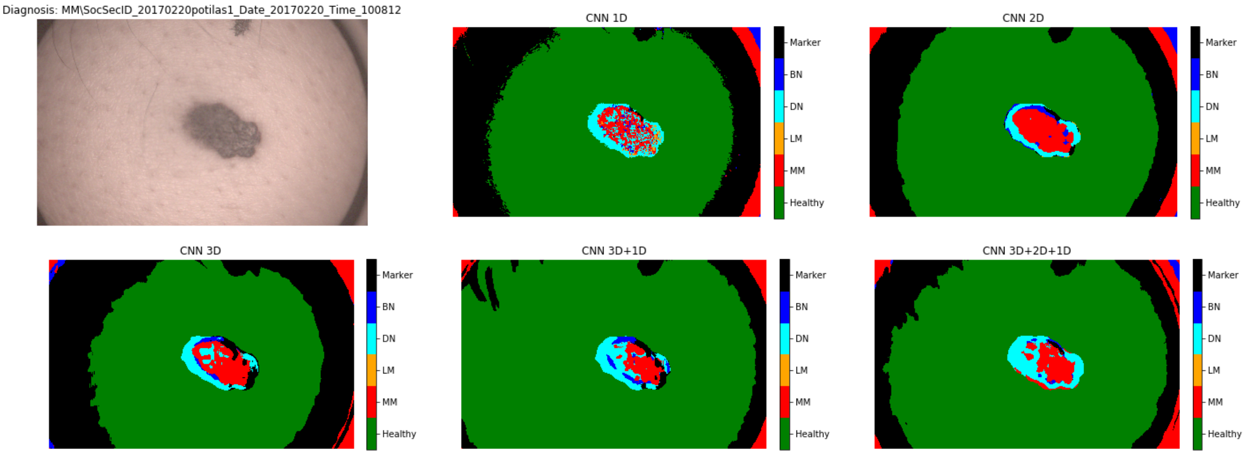

Diagnosis: MM\SocSecID_20170220potilas1_Date_20170220_Time_100812
CNN 1D
CNN 2D
CNN 3D
CNN 3D+1D
CNN 3D+2D+1D
Marker
BN
DN
LM
MM
Healthy

Figure 5. Classification results of five different classifiers for malignant melanoma.

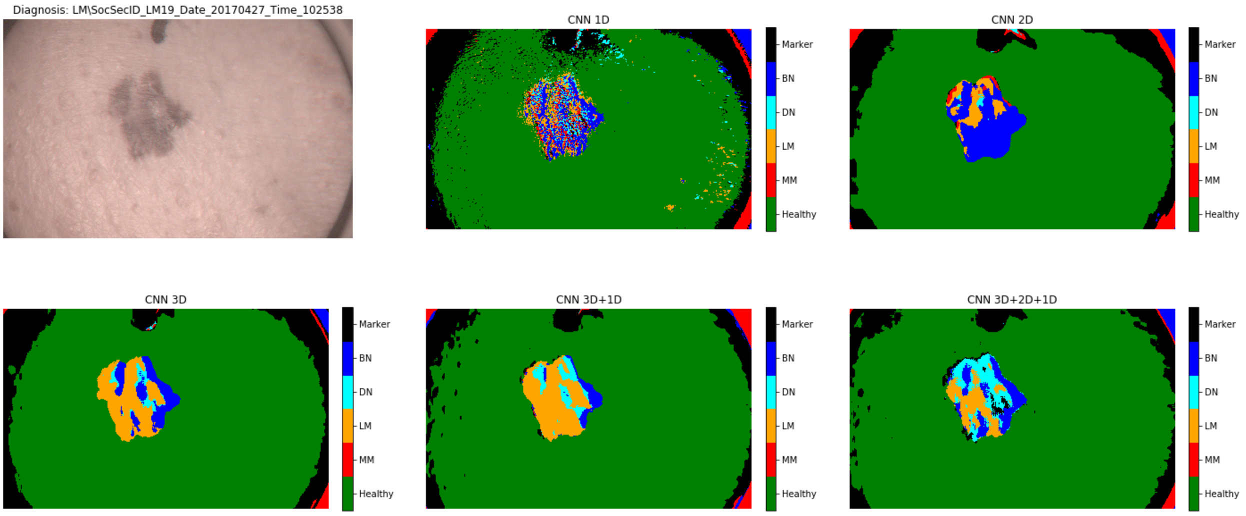


Figure 6. Classification results of five different classifiers for lentigo maligna.

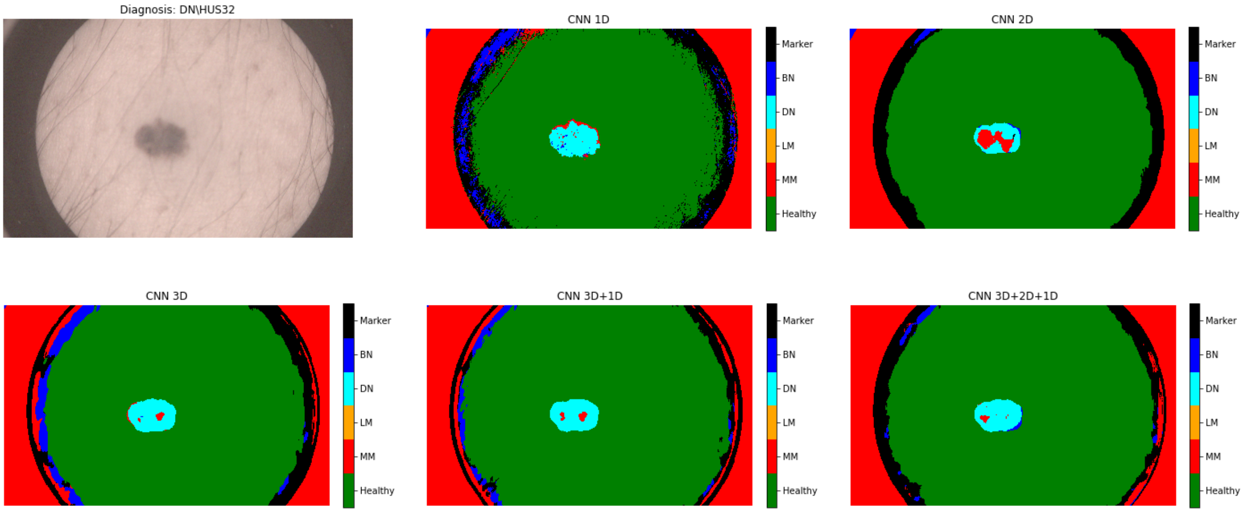


Figure 7. Classification results of five different classifiers for dysplatic nevus. Here all classifiers give false positive as a result. Even though majority of pixels are classified correctly, the end result will be false positive for whole lesion.

4. DISCUSSION

Shown results are promising. With all classifiers we achieved same positive prediction value (PPV) as clinicians. It is shown that utilisation of the spectral and spatial domain increases classification performance.

There is work to be done to gain higher specificity and PPV. We could play around with detection probabilities provided by the softmax layer and take some threshold probabilities, which would be concerned during classification (for example only classification results over 90% confident would be recognised). Or we could calculate which class has majority of pixels on lesion area. Unfortunately both of these approaches would actually decrease the sensitivity and the number of false negatives would rise.

Our study's first limitation comes from the small data set. Even thought we had over hundred million pixels at our disposal, we eventually had only 61 different lesions. This is a quite limited data set and more data is needed to develop and calculate a more robust and accurate neural network model. This would mean that we will need multi center studies, where patient data is gathered in several countries simultaneously. For example Finnish population is too small to produce enough patients to train enough general models.

Another limitation is that the ground truth labeling is based on histopathological diagnosis of whole lesion. There is a great possibility that a lesion can include several classes. Thus, our ground truth contains bias and this bias is also transferred to our training data. What we actually should do is that we should have several biopsied training points from each lesion so that we could use those spots in our training data. This would decrease bias in the training data, but it would also lead to reduced training data size.

Process of validating results and gathering training data should be similarly iterative as training of the neural network itself. When a hyperspectral imager and a classification model is used in a clinical study, we should take biopsies based on results. The spatial locations of these biopsies should be saved and the model should be updated using histopathological results of these studies afterwards.

The approach to use spectral and spatial domains seems feasible. Our next ideas are to add more features to the data. By modifying the illumination source we can take photogrammetric stereo images. From these images it is possible to calculate surface normals, a digital elevation model and skin's albedo as a function of wavelength. Each of these can be used as new features in cancer classification and delineation.

5. CONCLUSION

We have shown that use of spectral and spatial domain will increase classification performance of convolutional neural network. Our results show that with a relative small data set we are able to get same or slightly better positive prediction values as clinicians. This information was achieved by using a novel hyperspectral imager prototype in a clinical setup and train five different neural network models based on histopathological diagnoses. Because of the climate change proportion direct of sun radiation seems to grow, thus non-invasive automatic skin cancer detection and delineation systems will be needed even more in the future. These results are incremental steps towards this goal.

REFERENCES

[1] Eriksson, T. and Tinghög, G., "Societal cost of skin cancer in sweden in 2011," *Acta dermato-venereologica* **95**(3), 347–348 (2015).

[2] Heal, C. F., Raasch, B. A., Buettner, P., and Weedon, D., "Accuracy of clinical diagnosis of skin lesions," *British Journal of Dermatology* **159**(3), 661–668 (2008).

[3] Argenziano, G., Cerroni, L., Zalaudek, I., Staibano, S., Hofmann-Wellenhof, R., Arpaia, N., Bakos, R. M., Balme, B., Bandic, J., Bandelloni, R., et al., "Accuracy in melanoma detection: a 10-year multicenter survey," *Journal of the American Academy of Dermatology* **67**(1), 54–59 (2012).

[4] Neittaanmäki-Perttu, N., Grönroos, M., Jeskanen, L., Pölönen, I., Ranki, A., Saksela, O., and Snellman, E., "Delineating margins of lentigo maligna using a hyperspectral imaging system," *Acta dermato-venereologica* **95**(5), 549–552 (2015).

[5] Neittaanmäki, N., Salmivuori, M., Pölönen, I., Jeskanen, L., Ranki, A., Saksela, O., Snellman, E., and Gronroos, M., "Hyperspectral imaging in detecting dermal invasion in lentigo maligna melanoma," *Br J Dermatol* (2016).

[6] Krizhevsky, A., Sutskever, I., and Hinton, G. E., "Imagenet classification with deep convolutional neural networks," in [*Advances in neural information processing systems*], 1097–1105 (2012).

[7] Lawrence, S., Giles, C. L., Tsoi, A. C., and Back, A. D., "Face recognition: A convolutional neural-network approach," *IEEE transactions on neural networks* **8**(1), 98–113 (1997).

[8] Kim, Y., "Convolutional neural networks for sentence classification," *arXiv preprint arXiv:1408.5882* (2014).

[9] Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., and Thrun, S., "Dermatologist-level classification of skin cancer with deep neural networks," *Nature* **542**(7639), 115 (2017).

[10] Saari, H., Pölönen, I., Salo, H., Honkavaara, E., Hakala, T., Holmlund, C., Mäkynen, J., Mannila, R., Antila, T., and Akujärvi, A., "Miniaturized hyperspectral imager calibration and uav flight campaigns," in [*Sensors, Systems, and Next-Generation Satellites XVII*], **8889**, 88891O, International Society for Optics and Photonics (2013).

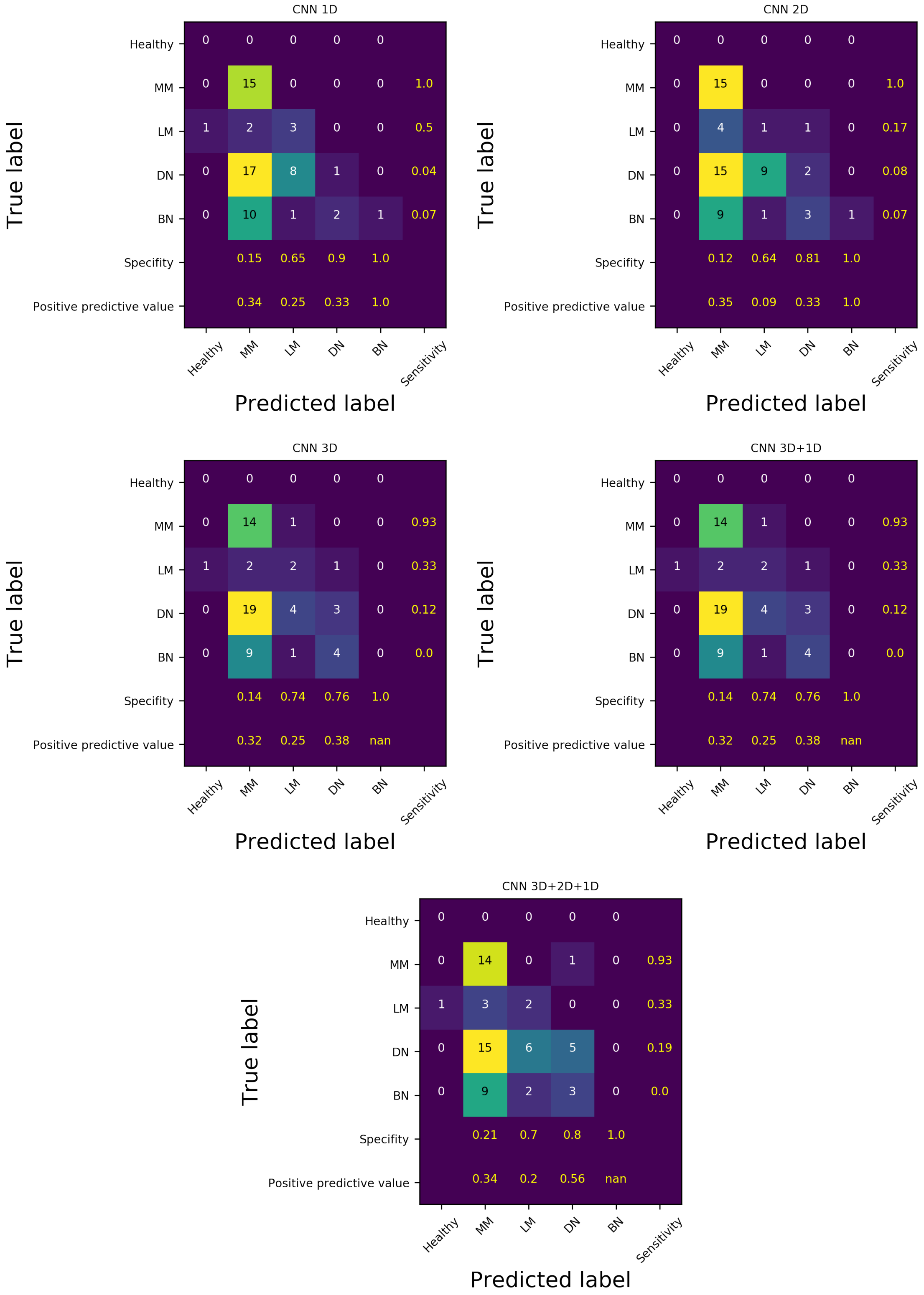


Figure 8. Confusion matrices for the different convolutional neural network models.

PII

TREE SPECIES IDENTIFICATION USING 3D SPECTRAL DATA AND 3D CONVOLUTIONAL NEURAL NETWORK

by

Ilkka Pölönen, Leevi Annala, **Samuli Rahkonen**, Olli Nevalainen, Eija Honkavaara, Sakari Tuominen, Niko Viljanen and Teemu Hakala 2019

WHISPERS 2018 : 9th Workshop on Hyperspectral Image and Signal Processing, Evolution in Remote Sensing

TREE SPECIES IDENTIFICATION USING 3D SPECTRAL DATA AND 3D CONVOLUTIONAL NEURAL NETWORK

Ilkka Pölönen[1]*, Leevi Annala*[1]*, Samuli Rahkonen*[1]*, Olli Nevalainen*[2]*, Eija Honkavaara*[2]*,*
Sakari Tuominen[3]*, Niko Viljanen*[2]*, Teemu Hakala*[2]

[1]Faculty of Information Technology, University of Jyväskylä, P.O. Box 35, FI-40014 Jyväskylä, Finland
[2]Finnish Geospatial Research Insititute, National Land Survey of Finland,
Geodeetinrinne 2, 02430 Masala, Finland
[3]Natural Resources Institute Finland, PL 2 00791 Helsinki, Finland

ABSTRACT

In this study we apply 3D convolutional neural network (CNN) for tree species identification. Study includes the three most common Finnish tree species. Study uses a relatively large high-resolution spectral data set, which contains also a digital surface model for the trees. Data has been gathered using an unmanned aerial vehicle, a framing hyperspectral imager and a regular RGB camera. Achieved classification results are promising by with overall accuracy of 96.2 % for the classification of the validation data set.

Index Terms— Tree species, spectral imaging, 3D, convolutional neural network, UAV

1. INTRODUCTION

This study is continuum for [1], where the individual tree detection and classification pipeline for the hyperspectral and point cloud data is clearly described. We are interested to see if deep learning methods could improve or simplify the data processing chain for identifying the species of individual trees.

There exists plenty of research concerning tree species identification, but it is mainly concentrated on large scale remote sensing, which uses forest stand and plot level data. For example in Scandinavia combination of airborne laser scanning and aerial images is used in forest inventory [2]. There are less studies and applications for the tree species identification from unmanned aerial vehicles (UAV) using hyperspectral sensors. If hyperspectral data has been used for tree species identification, the platform for data gathering has been manned aircraft or satellite.

As in [1], these remote sensing studies use quite traditional feature extraction and selection methods before classification. Deep learning methods have dramatically improved performance of pattern recognition [3]. Especially deep convolutional neural networks (CNN) have provided breakthroughs in image, video and audio processing. If we consider hyperspectral data, it seems that they should handle hyperspectral data combined with 3D data as well. There is currently increasing number of research, which applies CNN's and 3D CNN's to hyperspectral imager [4, 5].

In this paper we first test performance of 3D CNN for tree species classification. Neural networks has the nature of being a black box, that doesn't reveal how it has reasoned its results. However, while doing classification we can calculate saliency maps, which will give us hints on which parts of the input data are relevant for the CNN [6].

This paper has the following structure. First, in Section 2 we describe the used data set, its acquisition and preprocessing. Then the structure and functionality of the used 3D CNN is described. In Section 3, the results are presented and Section 4 includes the conclusion.

2. MATERIALS AND METHODS

2.1. Data gathering and preprocessing

The research data is the same as reported in [1]. The collected remote data was captured in Vesijako research forest area in the municipality of Padasjoki in southern Finland (approximately 61^o24'N and 25^o02'E). Area has been used for forestry research by Natural Resources Institute of Finland. The area contains experimental plots with different research setups. All the trees with the diameter of at least 50 mm at the breast-height were measured and estimated with various metrics, such as the tree species, diameter, height and volume. Locations of these trees were collected with GPS.

In total, 4142 trees were selected for further study. The data set contained three most common species of Finnish forests: scots pine (*Pinus sylvestris*, 2821 samples), norway spruce (*Picea abies*, 742 samples) and silver birch (*Betula bendula*, 579 samples). These selected trees were compared to aerial orthoimage mosaics to ensure that the GPS coordi-

This research has been co-financed by Finnish Funding Agency for Innovation Tekes (grants 2208/31/2013 and 1711/31/2016)

nates were in the centres of the treetops.

The used remote sensing data was a combination of two data modalities captured by the UAV remote sensing system, which belongs to Finnish Geospatial Research Institute. System consist of a Tarot 960 hexacopter and a Pixhawk autopilot. System is capable of carrying 3 kg payload at maximum. Average flying time of the system is 30 minutes. As a payload, we had a tunable Fabry-Pérot inteferometer based spectral imager (FPI) and an ordinary RGB camera, the Samsung NX1000 (RGB). Flying height from ground level varied between 83-94 meters.

The FPI imager captures raw data, which is processed to radiance based on the radiometric laboratory calibration [7]. The geometric imaging model was then determinated. The model includes both the interior and exterior orientations of the images. The digital surface model is calculated by dense image matching. Because of the slight variations between bands in the FPI camera, we had to apply registration of the spectral bands of FPI images. To make data cubes and further mosaics radiometrically homogenous, the radiometric imaging model has to be determined [8, 9]. The hyperspectral image mosaic is calculated after the radiometric model is applied to each cube. The detailed radiometric and geometric processing of the data set is explained in [1]. Finally, the spectral mosaics with 33 bands and digital surface model (DSM) both with 10 cm GSD are created.

For the tree species identification, 4×4 meter windows surrounding each treetop were extracted. The windows contained both DSMs as rasters and spectral cubes. For each treetop, the extracted DSMs were scaled by the minimum value of the whole DSM. The DSM and spectral cube for each treetop were concatenated in spectral axis to unified data cubes ($41 \times 41 \times 34$). In Finland there exist laser scanned nation wide ground surface elevation model, which is freely available. Thus, canopy surface model could have been calculated, but it isn't actually needed, because we are only using height of the treetops.

Figure 1 illustrates average treetops for each species. We can see that there are slight differences between the shapes. Pine's treetops are quite symmetric. Spruce's treetops are more of ellipses and aligned on north-west to south-east axis. Birch is more irregular, but its leaves and branches are towards south where the Sun shines.

Figure 2 represents how spectral distribution diverges to different wavelengths for each tree species. The line in the figure represents the average spectra for each treetop. Quite obvious differences can be found between birches and Nordic coniferous trees. Birches have stronger reflection in green and infrared regions. Birches have steeper spectrum at red edge area.

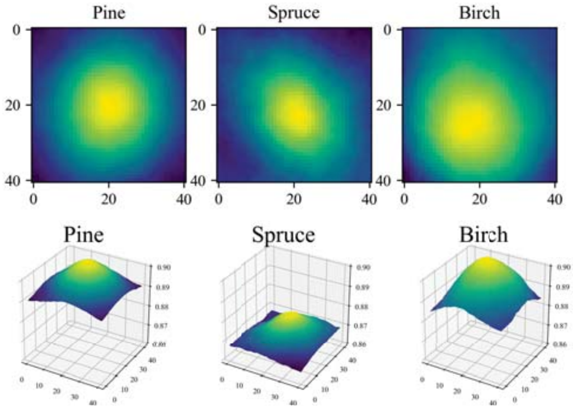


Fig. 1. Avarage shape of treetop for each tree species.

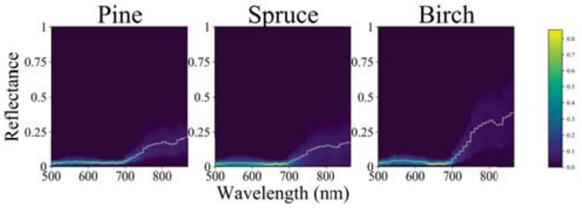


Fig. 2. Histogram spectra of each tree species. Black line is average spectrum.

2.2. Convolutional Neural Network

Originally CNN's were presented by LeCun and Bengio [10]. The idea was to tackle feature extraction and selection problem in fully connected feed-forward networks. The network uses a convolution matrices. Traditional neural network layers are usually based on consecutive dense (fully connected) neurons. In convolutional neural networks, there exists at least one convolution operation in the network. We applied quite simple structure to our CNN, using four types of layers: 3D convolutional, pooling, dropout and fully connected layers. Our network's structure is presented in Table 1.

In general, convolutional layers have trainable filters, which use convolution operations to extract features. In our implementation, the convolution layer uses activation called rectified linear unit (ReLU). ReLU has advantages of being efficient with non-linear relations and having less vanishing gradient problems during the network optimisation compared to other popular activation functions [11]. Pooling layers, which usually follow convolutional layers, are non-linear downsampling functions, which reduce dimensions of input data. Dropout layer is a regularization method for reducing overfitting in the neural network by introducing noise to the network. Flatten layer translates data to one dimensional stack.

Layer	Kernel / pool size or Activation	Output Shape	Parameters
Conv3D	(3,3,1) ReLU	(39, 39, 33, 64)	640
Conv3D	(3,3,3) ReLU	(37, 37, 31, 64)	110656
MaxPooling3D	(2,2,1)	(18, 18, 31, 64)	0
Conv3D	(3,3,3) ReLU	(16, 16, 29, 128)	221312
MaxPooling3D	(2,2,3)	(8, 8, 9, 128)	0
Conv3D	(3,3,3) ReLU	(6, 6, 7, 256)	884992
MaxPooling3D	(2,2,3)	(3, 3, 2, 256)	0
Flatten		(4608)	0
Dense	ReLU	(128)	589952
Dropout (0.25)		(128)	0
Dense	SoftMax	(3)	387
Total params:	1,807,939		
Trainable params:	1,807,939		
Non-trainable params:	0		

Table 1. Structure of our experimental CNN.

A dense layer is a fully connected layer, which consists of parallel neurons which are connected to all previous layer's outputs. Weights of the connections and activation functions determine which features are correlating with different tree species. The last dense layer is activated with *softmax* function, whose output is the final classification.

If the amount of data is limited, meaning that the number of training samples is low, then there is option to apply data augmentation. Basically this means that we will generate new training data from existing ones. In this study we fivefold our training data by using simple rotation and flipping operations. Selected training data was flipped both horizontally and vertically. Data was also rotated 90 degrees to left and right.

In machine learning structures like neural networks are so called "black box" solutions. We don't have clear vision how data is classified. It is reasonable to ask, is the classification based on real feature of wanted object or something secondary such as ground type in tree species recognition. Luckily there are methods to see where network is putting weight in classified data. It is possible to calculate gradient over layers from output to input. This way to get actually image, where areas with higher values contributes most to classification result. These maps are called *saliency maps*.

Stochastic gradient decent was used to tune weights between layers. We used categorical cross entropy as a loss function, which basically calculates cross entropy between categories probability distributions. Primary metric for model evaluation was accuracy

$$acc = \frac{TP + TN}{TP + FP + FN + TN},$$

where TP is true positive, TN is true negative, FP is false positive and FN is false negative classification result.

CNN's were trained by using IBM PowerAI platform which includes two Tesla V100-SXM2 16 GB GPU units. Tensorflow was used as a computational backend [12]. All machine learning phase coding was done using Python 3.6 and Keras library [13]. Saliency maps were calculated using Keras-vis library [14].

3. RESULTS

Altogether 3311 trees were randomly selected for the training of the 3D CNN. After data augmentation there was 16555 samples. Training was performed with batch size 128 and with 100 epocs. Training took two and half hours (approx. 88 seconds/epoch). Results were validated with 831 samples, which weren't included in training set.

Figure 3 shows that accuracy of trained model is relatively high. It seems that we can with quite large confidence identify tree species from each other. Overall accuracy for classification of validation set was 96.2%, which is higher with earlier results achieved in [1]. Producer accuracies were for each tree species were 96.2% (Pine), 86.6 % (Spruce) and 98.2 % (Birch). Respectively users accuracies were 96.3 %, 83.8 % and 95.7 %.

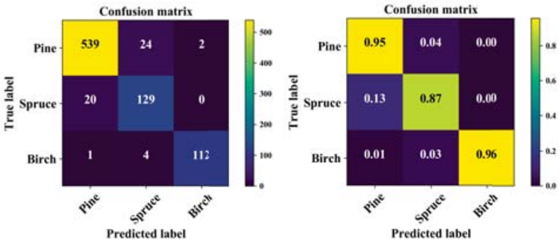


Fig. 3. Confusion matrices show good separation between tree species.

Figure 4 is presenting average saliency maps in spatial domain over all input bands of validation data. It seems that most of the important features are handling data surrounding tree top. This is shown more clear in the figure 5, where figure's 4 maps are rendered over validation sets average 3D treetops. Thus, we can be quite confident that, at least in spatial domain, tree top's shape is relevant feature in classification.

In spectral domain most characterising features seems to be located between wavelengths from 600 to 720 nm. Figure 6 presents average salience in each spectral band. It can seen that there is differences between tree species. For example birch has lower saliency in 560 nm and higher in 700 nm than coniferous trees.

If we consider individual trees, it seems that classifying is working quite efficiently. In figure 7 there is one tree of each species from the validation set. It can be seen that for example pine in this case doesn't have very clear treetop, but classifier is able to find one and saliency map seems to confirm the result.

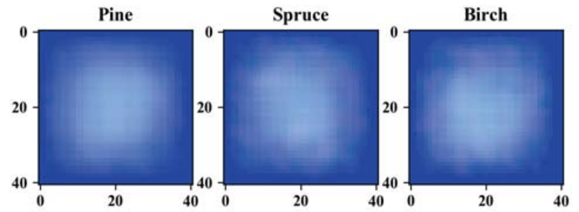


Fig. 4. Average saliency maps in spatial domain over all input bands of validation data for each tree species. Brighter pixel indicates that band is probably more meaningful in classification.

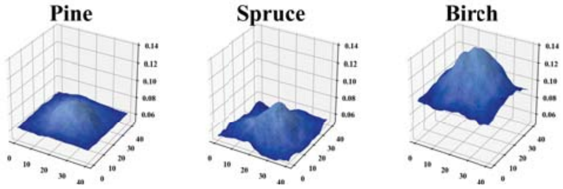


Fig. 5. Here figure's 4 saliency maps are rendered over validation set's average 3D treetops. It can be seen that maps surround quite well treetops.

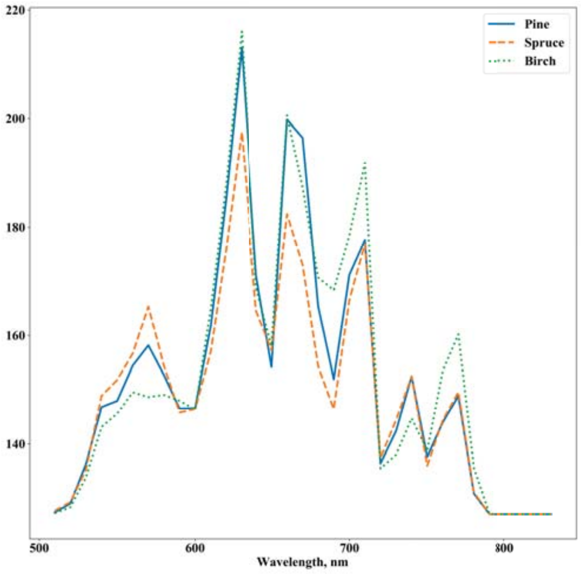


Fig. 6. Average saliencies of spectral domain for each tree species. Higher value indicates that band is probably more meaningful in classification.

4. CONCLUSIONS

In this paper we demonstrate how 3D hyperspectral data can be analysed using 3D convolutional neural networks. As a concluded result we can see that even with quite simple 3D CNN, it is possible to create network, which has good capability to classify single trees based on their shape and spectral features.

In classical machine learning one of the most time consuming thing for data analysist has been feature extraction and selection. In case of convolutional neural network this phase is now automated. After preprocessing there is quite limited amount of things to do, if you want to utilize trained network. Network training itself is time consuming, but before hand trained network can deliver results almost in realtime. In our case training took two and half hours.

Compared to earlier work [1] we actually used all captured test areas. In original paper one area was left behind, because of the poor quality of image block. Based on that, our results seems to show that trained 3D CNN is actually more robust as a classifier than methods used in previous study.

It is obvious that more studies is needed. Used network structure is one of the most simple ones. With more sophisticated structures it might be possible to improve learning results. One of the tested things in the future is, how general trained model actually is. If we have another data set, can we have similar classification results? We used quite limited amount of data augmentation. Even tough overfitting wasn't observed based loss and accuracy curves during the training, it would be useful to do more cross-validation within data.

One potential research question is that how many bands and what GSD is needed, if we want to gain similar results. Our next steps include more augmented data to training such as scaling, adding noise, chancing lightness and adding more rotation to see if we could detect trees also with lower resolution.

The used data set has more parameters for single trees (height, estimated volume, etc..) and there is also 300 fixed radius (9 m) sample plots, which have been used for area based forest inventory. In near future we will also test how well 3D CNN approach is able to estimate these parameters.

Our consortium has ongoing research project where our aim is to produce real time processing for the DSM and hyperspectral mosaics. This combined with pre-trained CNN classifier, could be significant tool to provide forest tree identification and parameter estimation without wasting time on massive preprocessing.

5. REFERENCES

[1] Olli Nevalainen, Eija Honkavaara, Sakari Tuominen, Niko Viljanen, Teemu Hakala, Xiaowei Yu, Juha Hyyppä, Heikki Saari, Ilkka Pölönen, Nilton N Imai, et al., "Individual tree detection and classification with uav-based photogrammetric point clouds and hy-

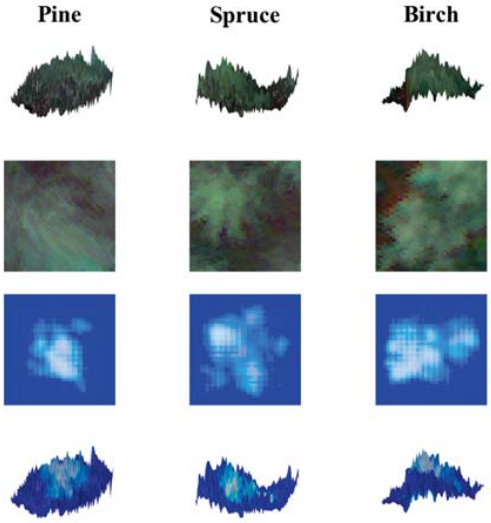


Fig. 7. Comparison individual trees. CNN is capable to detect trees surprisingly well.

perspectral imaging," *Remote Sensing*, vol. 9, no. 3, pp. 185, 2017.

[2] Erik Næsset, "Predicting forest stand characteristics with airborne scanning laser using a practical two-stage procedure and field data," *Remote sensing of environment*, vol. 80, no. 1, pp. 88–99, 2002.

[3] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[4] Sungbin Choi, "Plant identification with deep convolutional neural network: Snumedinfo at lifeclef plant identification task 2015.," in *CLEF (Working Notes)*, 2015.

[5] Luiz G Hafemann, Luiz S Oliveira, and Paulo Cavalin, "Forest species recognition using deep convolutional neural networks," in *Pattern Recognition (ICPR), 2014 22nd International Conference on*. IEEE, 2014, pp. 1103–1107.

[6] Mariusz Bojarski, Philip Yeres, Anna Choromanska, Krzysztof Choromanski, Bernhard Firner, Lawrence Jackel, and Urs Muller, "Explaining how a deep neural network trained with end-to-end learning steers a car," *arXiv preprint arXiv:1704.07911*, 2017.

[7] Heikki Saari, Ilkka Pölönen, Heikki Salo, Eija Honkavaara, Teemu Hakala, Christer Holmlund, Jussi Mäkynen, Rami Mannila, Tapani Antila, and Altti Akujärvi, "Miniaturized hyperspectral imager calibration and uav flight campaigns," 2013, vol. 8889, pp. 88891O–88891O–12.

[8] L. Markelin, E. Honkavaara, R. Näsi, K. Nurminen, and T. Hakala, "Geometric processing workflow for vertical and oblique hyperspectral frame images collected using uav," *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XL-3, pp. 205–210, 2014.

[9] Eija Honkavaara, Heikki Saari, Jere Kaivosoja, Ilkka Pölönen, Teemu Hakala, Paula Litkey, Jussi Mäkynen, and Liisa Pesonen, "Processing and assessment of spectrometric, stereoscopic imagery collected using a lightweight uav spectral camera for precision agriculture," *Remote Sensing*, vol. 5, no. 10, pp. 5006–5039, 2013.

[10] Yann LeCun and Yoshua Bengio, "Convolutional networks for images, speech, and time-series," 1995.

[11] Xavier Glorot, Antoine Bordes, and Yoshua Bengio, "Deep sparse rectifier neural networks," in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, 2011, pp. 315–323.

[12] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, Software available from tensorflow.org.

[13] François Chollet et al., "Keras," 2015.

[14] Raghavendra Kotikalapudi and contributors, "keras-vis," https://github.com/raghakot/keras-vis, 2017.

PIII

MULTILABEL SEGMENTATION OF CANCER CELL CULTURE ON VASCULAR STRUCTURES WITH DEEP NEURAL NETWORKS

by

Samuli Rahkonen, Emilia Koskinen, Ilkka Pölönen, Tuula Heinonen, Timo Ylikomi, Sami Äyrämö and Matti A. Eskelinen 2020

Journal of Medical Imaging, SPIE

Multilabel segmentation of cancer cell culture on vascular structures with deep neural networks

Samuli Rahkonen,[a,*] Emilia Koskinen,[b] Ilkka Pölönen,[a] Tuula Heinonen,[b] Timo Ylikomi,[b] Sami Äyrämö,[a] and Matti A. Eskelinen[a]

[a]University of Jyväskylä, Faculty of Information Technology, Jyväskylä, Finland
[b]Tampere University, Faculty of Medicine and Health Technology, Finnish Centre for Alternative Methods, Tampere, Finland

Abstract. New increasingly complex *in vitro* cancer cell models are being developed. These new models seem to represent the cell behavior *in vivo* more accurately and have better physiological relevance than prior models. An efficient testing method for selecting the most optimal drug treatment does not exist to date. One proposed solution to the problem involves isolation of cancer cells from the patients' cancer tissue, after which they are exposed to potential drugs alone or in combinations to find the most optimal medication. To achieve this goal, methods that can efficiently quantify and analyze changes in tested cell are needed. Our study aimed to detect and segment cells and structures from cancer cell cultures grown on vascular structures in phase-contrast microscope images using U-Net neural networks to enable future drug efficacy assessments. We cultivated prostate carcinoma cell lines PC3 and LNCaP on the top of a matrix containing vascular structures. The cells were imaged with a Cell-IQ phase-contrast microscope. Automatic analysis of microscope images could assess the efficacy of tested drugs. The dataset included 36 RGB images and ground-truth segmentations with mutually not exclusive classes. The used method could distinguish vascular structures, cells, spheroids, and cell matter around spheroids in the test images. Some invasive spikes were also detected, but the method could not distinguish the invasive cells in the test images. © *The Authors. Published by SPIE under a Creative Commons Attribution 4.0 Unported License. Distribution or reproduction of this work in whole or in part requires full attribution of the original publication, including its DOI.* [DOI: 10.1117/1.JMI.7.2.024001]

Keywords: neural network; segmentation; cancer; *in vitro*; microscopy.

Paper 19073RRR received Mar. 18, 2019; accepted for publication Mar. 23, 2020; published online Apr. 7, 2020; corrected Apr. 13, 2020.

1 Introduction

In many cases, chemotherapeutic anticancer therapies' contribution to life extension is low, and serious adverse effects are common.[1,2] Also the cost of cancer treatments is extremely high and is increasing constantly.[2]

To improve the clinical benefits of treatments, there is a need to find new ways to treat cancer and to investigate which drugs or drug combinations are the most effective for individual cancers and patients. This could be achieved by developing new *in vitro* models that are able to reliably predict the effect of drugs *in vivo*. To get more relevant and reliable information from *in vitro* techniques, *in vitro* cancer research has been increasingly interested in alternative models that better mimic the tumor environment *in vivo*.[3] These new models are often complex and have many parameters that need to be considered simultaneously. Therefore, there is a need for computational methods to recognize and quantify these new parameters and handle the massive amount of data with high throughput.

In this study, we cultivated prostate carcinoma cell lines PC3 and LNCaP on top of a matrix with vascular structures. We were able to identify some characteristics, such as structures and appendages, which are thought to represent invasiveness and possible metastatic potential. A phase microscope was used for taking the images. The PC3 cells lined up with the vascular structures and could be seen as lined-up matrix-covering networks at later time points. The LNCaP

*Address all correspondence to Samuli Rahkonen, E-mail: samuli.rahkonen@jyu.fi

cells grew in spheroids that in the last imaging days showed spike-like formation growing from their edges with cells and cell matter surrounding the spheroids. Changes in spheroids were demonstrated to correlate with gene expression patterns in studies by other research groups.[4,5]

Segmentation is a technique to extract additional features from images, which can be used for further analysis. Image segmentation means classifying images at the pixel level. In this case, the microscopy images of cell cultures could be classified per pixel to different cells and structures.

The limitation of the used imaging method is that it is difficult to distinguish different cells from each other. Staining the cells would make identification easier, but the available staining options could damage or even kill the cells in addition to increasing expenses and labor. Disturbances to cells change their movements and surface proteins enough to influence the results. Also, the fluorescence stains might have unknown and unwanted interactions with cancer drugs. This prevents the use of the common solution in literature in which fluorescence markers would be used as segmentation labels.[6] Fluorescence technology requires excitation, which also needs to be avoided due to possible interference with later applied study chemicals and drugs.[7]

Further feature engineering could reveal new information on how different cells and structures interact. Possible features are, for example, the morphological features, such as the area, perimeter, and diameters of different cells.[6] Also using distances between different cells and structures could be used to identify spatial and functional relationships.[8] For example, Ref. 9 Ahonen et al. measured how different drug treatments can have an effect on the area and roundness of cells. Assessment of the drug efficacy could be achieved similarly after the structures were segmented first.

Six classes of structures, which were thought to include meaningful data for testing the drug efficacy, were identified in the images. Since the structures are semitransparent and may overlap, we are considering a multilabel problem with mutually not exclusive classes. Therefore, the proposed solution approaches the problem by carefully constructing datasets for training multiple specialized U-Net neural networks[10] to detect the classes from microscopy images.

2 Related Work

Segmentation is used widely in medical image analysis. Many past applications are based on traditional image processing algorithms to produce segmentations tailored to the problem. For example, Ref. 9 used the local entropy filter and Watershed algorithms for thresholding cell culture images and a support vector machine for segmentation. The structures were extracted from the image background and separated from each other. The texture features were engineered based on, e.g., the local binary pattern histogram and Haralick features.

Deep neural networks (DNNs) do not need such feature engineering for classification as the features are automatically learned during the training. DNNs have been applied in many different medical image segmentation tasks. For example, Ref. 11 applied neural networks in segmenting chest radiographs and Ref. 12 used a DNN for predicting fluorescent labels from transmitted-light z stack microscopy images.

Reference 6 conducted a study in which images of fluorescently labeled tissue samples were segmented by extracting image patches around each pixel and feeding them to a deep convolutional neural network for classification. The cell type with the maximum predicted probability was assigned to each pixel so that each pixel was classified to a single class. This kind of technique is known as the sliding window.

Another approach for segmentation is the encoder–decoder network. Original U-Net is one implementation of this kind of network, which takes an image as an input and creates a segmentation for the whole image.[10] U-Net has been used for many different biological image segmentation tasks. Reference 10 used their DNN (U-Net) in neuronal structures in electron microscopic recording segmentation. Reference 13 demonstrated how a three-dimensional U-Net architecture can be used in delineating radiosensitive organs in head and neck.

The latest DeepLabv3 uses spatial pyramid pooling with different convolutional grid scales to extract rich semantic features. Convolution operations use strides to support large image sizes in situations in which there are scarce memory and computational resources available. It uses a simple decoder module similar to U-Net to capture sharp object boundaries in the segmentation.[14]

A newer approach is the generative adversarial network, which has also been used in segmentation by conditioning the image generation, but they have been seen to produce hallucinated structures that do not exist in the original input images.[15]

It can be difficult to infer information about mutually not exclusive classes.[16,17] Segmenting this kind of multilabel images is less common. The class imbalance is also a problem. One way to handle the class imbalance is by oversampling and undersampling the training data. One could also modify the loss function by weighting the classes with varying amounts of data.[11,16] The problem has been approached by ensembling many models or a single end-to-end DNN model.[16]

3 Methods

3.1 *Experimental Setup*

Cryopreserved PC3 and LNCaP cells were obtained from the University of Turku. The cells were cultured in flasks for 3 days and then were seeded at a density of 5000 cells/well on the top of *in vitro* vascular structures in 96-well plates. Vascular structures were formed as described in Ref. 18 and subsequently decellularized by the method modified from Ref. 19. Cells were maintained in RPMI 1640 supplemented with 1% L-glutamine, 1% penicillin/streptomycin, and 10% fetal bovine serum (Gibco, Thermo Fisher Scientific) in a humidified incubator at 37°C and 5% CO_2 level. The imaged cells were not exposed to any drugs. The culture period was 14 days in total based on previous experiments.

The images were obtained by acquiring a 2 × 2 image grid from each well on days 4, 7, 11, and 14 after seeding using noninvasive imaging technology with a Cell-IQ phase-contrast microscope (CM technologies Oy. Tampere, Finland). The time points were chosen to allow the monitoring of the development of the structures seen in the phase-contrast images. In addition, the use of multiple time points allows the different cell types to form their growth patterns, regardless of their different proliferation rates. LNCaP cells are known to proliferate more slowly than PC3 cells. LNCaP cells reached their final growth pattern at the last imaging days. On the contrary, PC3 cells outgrew their wells quickly.

All computations were run on an IBM PowerNV 8335-GTG with two Tesla V100-SXM2 GPUs and 569 GB RAM. The neural networks were created with Python 3.6 and TensorFlow 1.10.[20]

3.2 *Dataset*

The study used 36 full color images (8-bit RGB, resolution 1392 × 1040) from two prostate-cancer-derived cell lines (PC3 and LNCaP), which were grown on top of vascular structures. PC3 and LNCaP cells were used because they form structures that have defined edges, which can be distinguished by visual inspection.

The corresponding ground-truth segmentations were created manually by a professional. The ground-truth targets had six classes and the background. The classes could overlap each other; therefore, the images were divided into separate ground-truth images with one target class per image (and the background). Each neural network did binary segmentation to each of these classes. Figure 1 shows three example images from the dataset. All classes are listed below with their corresponding short labels:

- background in green;
- noninvasively growing cells in red (S);
- invasive cells in white (I);
- invasive spikes in black (IP);
- vascular structures in blue (P);
- spheroids LNCaP cells in light blue (O); and
- cell matter around spheroids in yellow (R).

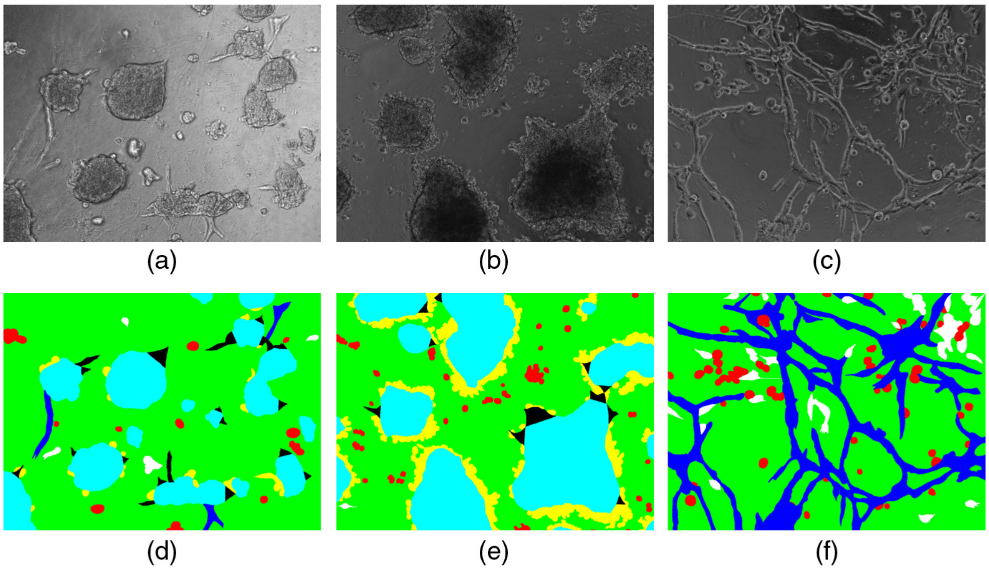


Fig. 1 Example training images and their corresponding unprocessed ground-truth images. Multilabel targets are stacked on top of each other. (a) and (b) Images from LNCaP cell line; (c) image from PC3.

The dataset had some nonuniform markings; thus, correct class information was not available for all images, and some had to be abandoned.

The cells were grown on top of the vascular structures and the invasive and noninvasive cells could overlap each other. This makes the classes mutually not excluding. From the images it could be seen that class P overlapped the most with classes S and I, and I could overlap S. Some images were mostly "empty," including only a few objects of interest. Some were densely populated with large areas of different structures.

The formation of spheroids, invasive spikes, and the cell matter around them were only observed with LNCaP cells, and distinct vascular networks were mostly observed in PC3 cells. We used images from both cell lines to train neural networks to ensure a sufficient amount of data for training and testing the neural networks.

This limited the number of available images for this study as manual annotation of large images is very time-consuming. Time constraints in the project were the main reason that there are not more images available. The lack of images was compensated for by other techniques in preprocessing, network training, and using multiple networks.

3.3 *Preprocessing*

Single images were too large to be used with U-Net and available GPU memory. Therefore, the images were split into 512×512 subimages. By splitting the images, we obtained a more balanced class distribution.

A total of 34 full color images and their ground truths were used in the training. Only two full color images could be used in testing because the dataset was small and the number of classes was relatively large.

To create a more balanced training dataset, each image was scanned through and the subimages, including a specific label, were identified. The resulting subimages contained only the target class and background, as depicted in Fig. 2.

Among the group of all subimages for an image with a specific label, at most 20 subimages, including pixels belonging to the class, were selected. These images were the positive samples. The same number of negative samples without the targeted class pixels was also selected. This was repeated for each class. The subimages could overlap each other. The method created many differently aligned views of the same objects of interest. Oversampling or undersampling

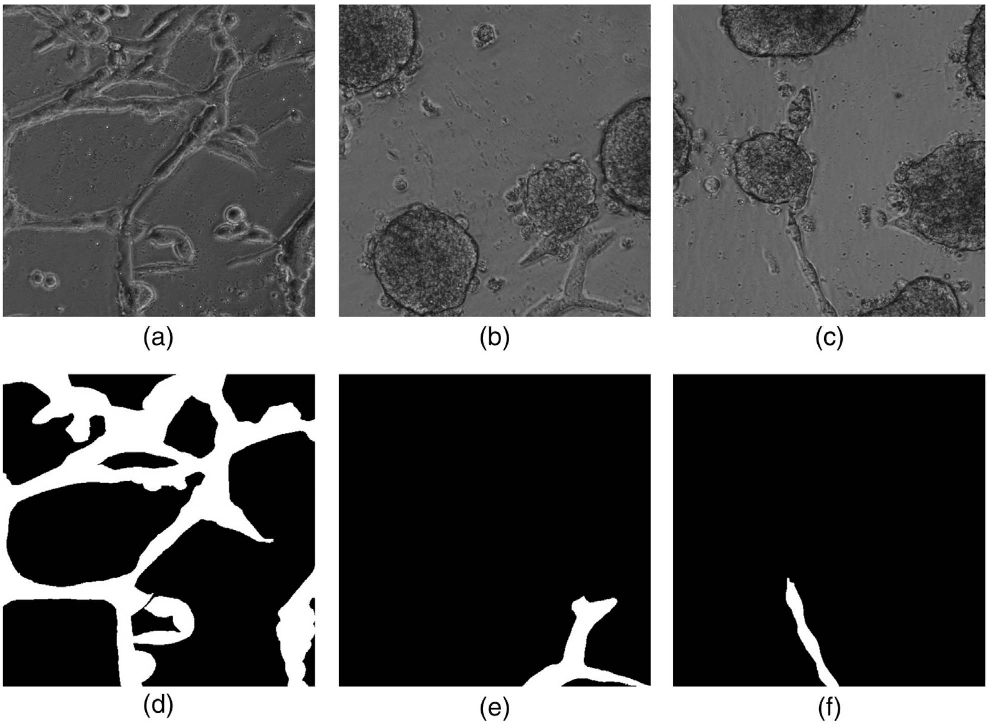


Fig. 2 Example training image patches extracted from full images and their corresponding ground-truth images (class P). (a) Image from PC3 cell line; (b) and (c) images from LNCaP.

Table 1 Numbers of samples used in training.

	P	S	IP	R	O	I
Training samples	1100	660	1239	1020	920	762

between classes was not used because each class was predicted by a separately trained neural network. The numbers of subimages for each class are listed in Table 1.

3.4 *Network Architecture*

A set of U-Net neural networks was the chosen method. U-Net is a neural network designed for biological image segmentation and for working with small training datasets. Rather than classifying image pixels one by one using a sliding window technique, U-Net inputs and outputs the whole image. This makes the training more efficient. U-Net should be a good choice for the cases in which there are not many training samples available.[10]

U-Net is constructed of convolutional layers and pooling layers that first decrease the resolution of the output. They are followed by upsampling convolutional layers that also concatenate the output in the feature channel axis with the results of the previous layers.

U-Net has a softmax activation function at its last layer and as such is not applicable for classifying mutually not excluding classes. A logistic function could have been used, but the assumption was that one specialized network for a single class would provide more accurate results. Also, the lack of data and the class imbalance encouraged us to train multiple networks, each with specifically selected training images. As a result, each network had two outputs: one for the individual class being predicted and one for everything else.

The created implementation is based on a heavily modified version of U-Net TensorFlow implementation created by Ref. 21.

3.5 *Training*

Based on the earlier tests, cells form two kinds of distinct growth patterns in this *in vitro* method, regardless of the cancer cell line used. Images from both LNCaP and PC3 cell lines were used for training because both cell lines produced structures with similar properties. Also, to increase the number of samples from which to draw subimages for the training, the data from both cell lines were pooled.

To fit the data into the GPU's memory, the images were first read into the memory of the server and were then fed to the GPU in batches of one or three images (the batch size was one for class P). Additional training examples were generated by augmentation. The data pipeline both rotated and flipped the images horizontally and vertically randomly. The brightness of the images was also randomly changed.

The learning rate, optimizer, and loss function were selected manually based on the literature and experience. Adam[22] optimizer was used. The weights of the networks were initialized with Glorot and Bengio.[23] The learning rate was 1×10^{-5} and set to decay following a staircase function, decreasing every 10,000th global TensorFlow step by the decay rate of 0.9.

The loss function was defined based on Ref. 11 with a regularization term. The purpose of the loss function is to quantify the difference between predictions and ground truths for steering the training of the network. It is defined as

$$C(I, G_I) = \sum_{l \in \mathcal{L}} r_{l,\mathcal{K}} d(I, G_I) + \lambda \sum_{w} w^2, \tag{1}$$

where the first term calculates the summed loss of the prediction compared with the ground truth, weighted by the class frequency. Here I is the set of images, $\mathcal{K}$ is the index of image batch, G_I denotes the ground-truth classes of the images, and d is the commonly used softmax cross entropy for the two outputs of the last layer. The outputs are the targeted class and the background. The class is denoted by l belonging to set of classes $\mathcal{L}$, whose size is two for each of the neural networks.

The last term is the L2 regularization over the trainable weights (w). Its purpose is to avoid overfitting the data. Weight factor λ was set to 0.001 because it seemed to reduce the variation of the loss during the training. Pixel-wise batch weighting was utilized by the weighting coefficient $r_{l,\mathcal{K}}$:

$$r_{l,\mathcal{K}} = \frac{c_{\mathcal{K}}}{c_{l,\mathcal{K}}}, \tag{2}$$

where $c_{l,\mathcal{K}}$ is the number of pixels in the batch $\mathcal{K}$ belonging to class $l \in \mathcal{L}$.[11]

The stopping criterion of the training was fixed to 1000 epochs (roughly 5 days of computation time) per neural network. The losses and accuracies of the predictions of the training and test sets were observed with TensorBoard to make sure the models do not overfit during the training. TensorBoard is a tool that can be used to visualize model metrics, such as loss and accuracy during the training. The losses of both training and test data were observed to converge. Because of the lack of data, we did not have a separate validation dataset available. Therefore, the model weights were not selected using the lowest loss for the test data. Optimizing the model to the test data, which is used for calculating the final performance metrics, would introduce bias to the results. The model weights of the 1000th epoch were selected.

3.6 *Test Image Processing*

The full test images had to be split into subimages to be used with the trained networks. For a better view of results, the resulting segmentations had to be postprocessed by combining them back into full-sized images.

Splitting the image in a simple grid was not appropriate because the segmentation created clearly visible artifacts at the edges of subimages. Therefore, a grid of overlapping subimages (30 linearly spaced coordinates in both x and y axes) inside the original image were selected. This resulted in 900 subimages.

All of these subimages were run through the neural network and combined by pixel-wise averaging. The resulting full images show some artifacts, especially at the edges where the pixels are averaged the least.

Other techniques for testing include cross validation and leave-one-out, but they were not used because training the networks for many dataset splits would have been too time-consuming.

3.7 *Performance Metrics*

We used sensitivity (true positive rate or recall), specificity (true negative rate), Dice score (DSC), and area under curve (AUC) metrics. True positives are the predictions that are correctly classified as positive, false positives are wrongly predicted as positives, true negatives are correctly classified as negatives, and false negatives are wrongly predicted as negatives. Manual thresholding was used for calculating these metrics.

Sensitivity corresponds to the proportion of positive data points that are correctly predicted as positive with respect to all positive data points. In other words, higher sensitivity means that fewer positive data points are missed.

Correspondingly, the specificity is the proportion of negative data points that are correctly predicted as negative.

DSC (also known as F1 score) is a similarity index for measuring spatial overlapping of manual ground-truth segmentations and the predictions of automatic methods. DSC ranges from 0 to 1, where 0 indicates no overlap and 1 is a complete overlap.[24]

Receiver operating characteristic (ROC) curve is a method used for assessing the performance of classification algorithms. It is widely used in medical diagnostics. The AUC is the integral of ROC, and one interpretation of it is the probability that the classifier will rank a randomly chosen positive example higher than a randomly chosen negative example. A correct classifier has an ROC above the diagonal and an AUC larger than 0.5.[25]

4 Results

Tables 2 and 3 contain the class-wise sensitivity, specificity, and AUC scores for both test images 1 and 2. The results have been calculated from thresholded predictions, where the threshold (0.5) was selected manually based on experience. If we had enough data for a validation dataset, that could have been used with ROCs to select a more optimal operating point. Using the test data to select the thresholds would introduce bias to the results.

Visualizations of the predictions and their classification errors for test images are illustrated in Sec. 7. Figures 4 and 5 are for image 1. For image 2, the visualizations are shown in Figs. 6 and 7.

Table 2 Sensitivity, specificity, AUC with 95% confidence interval, and DSC of image 1. The operating point (threshold) of the ROC was 0:5.

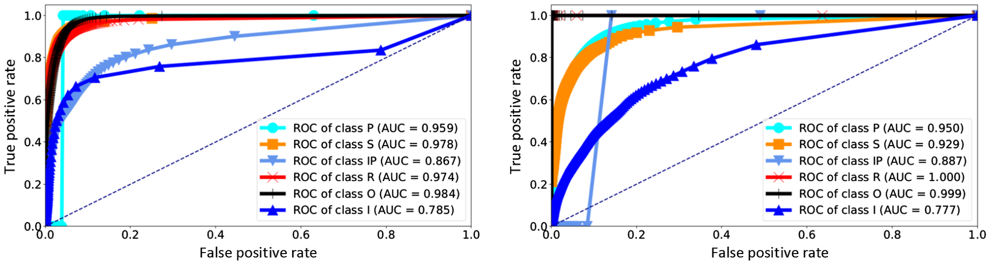
Class	Sens.	Spec.	AUC	DSC
P[a]	0.0	0.998	0.959 [0.959, 0.959]	0.0
S	0.61	0.994	0.978 [0.977, 0.979]	0.533
IP	0.346	0.986	0.867 [0.863, 0.869]	0.261
R	0.893	0.957	0.974 [0.974, 0.975]	0.822
O	0.703	0.983	0.984 [0.983, 0.984]	0.807
I	0.028	0.999	0.785 [0.768, 0.802]	0.03

[a] Class does not appear in the image.

Table 3 Sensitivity, specificity, AUC with 95% confidence interval, and DSC of image 2. The operating point (threshold) of the ROC was 0.5.

Class	Sens.	Spec.	AUC	DSC
P	0.642	0.968	0.949 [0.949, 0.949]	0.741
S	0.591	0.976	0.929 [0.928, 0.930]	0.578
IP[a]	0.0	1.0	0.887 [0.886, 0.887]	0.0
R[a]	0.0	1.0	0.999 [0.999, 0.999]	0.0
O[a]	0.0	1.0	0.999 [0.999, 0.999]	0.0
I	0.275	0.957	0.777 [0.775, 0.779]	0.245

[a] Class does not appear in the image.



(a) (b)

Fig. 3 ROCs for the test images (a) 1 and (b) 2.

4.1 *Overall Results*

The AUC scores were relatively high and ranged between 0.777 and 0.984, which implies that the classifiers' predictions were more often correct than false. The ROCs are shown in Figs. 3(a) and 3(b).

Classes R, O, P, and S all had high sensitivities (0.591 to 0.893). By contrast, classes I and IP had very low sensitivities (0.028 to 0.346). Class I was mainly misclassified as classes P and S.

The specificity was quite high with all of the classes because the models were trained with only two classes: the class in question and the background. Most of the images were filled with the empty background, which led to high specificity.

DSC had high overall variability between classes (0.03 to 0.822). The variability between images could be assessed only with classes S (0.533 to 0.578) and I (0.03 to 0.245). DSC scores were reasonable for P, S, R, and O, as shown in Tables 2 and 3. Classes I and IP had low DSC scores, due to many false negatives.

The models did not make false positive predictions in images that did not include any objects of the targeted class. Therefore, class P had 0.0 sensitivity in image 1. Classes IP, R, and O had 0.0 sensitivity in image 2. This can be seen as very steep ROCs.

4.2 *Class-Specific Results*

4.2.1 *Vascular structures*

For class P, there was one test image. The predictions did not include many false positives with the specificity of 0.968. Nevertheless, it did not classify the structures correctly where the area was crowded with unclear cells and structures. This affected the sensitivity (0.642).

4.2.2 *Invasive cells*

Class I's bad performance shows in the DSC and AUC scores, which were the lowest among all of the classes. Class P was often misclassified as class I, which can be especially seen in Fig. 7(i). The classes share similar properties in their shapes. Class I was not detected in either of the images. It was misclassified as vascular structures (P) and noninvasively growing cells (S), which resulted in many false positives and false negatives.

4.2.3 *Invasive spikes*

Invasive spikes had low sensitivity (0.346) and DSC (0.261). The model made some true positive predictions but also many false positive predictions on the surfaces of the spheroids, as seen in Figs. 4(l) and 4(o).

4.2.4 *Noninvasive cells*

Class S cells had 0.591 sensitivity with image 1 and 0.61 with image 2. In image 1, the problem was distinguishing separate cells from the cell matter around spheroids (class R).

4.2.5 *Spheroids*

Class O was probably the most recognizable class with distinctive large round shapes. It had good scores with 0.703 for sensitivity and 0.807 for DSC, but suffered from the selected threshold value, which truncated large areas from some of the spheroids.

4.2.6 *Cell matter around spheroids*

Class R had the highest sensitivity among all classes with 0.893. The AUC (0.974) and DSC (0.822) were both good. Figure 5(j) shows how cell matter is detected mostly correctly, but it had problems with the tight area between spheroids, as it was classified to class O.

5 Discussion

Classes R, O, P, and S had very distinctive shapes and clearly defined edges, which led to high sensitivities. Overall, the most evident structures were classified mostly correctly. Classes I and IP had very low sensitivities (0.028 to 0.346), and they were mostly misclassified as other classes.

Some of the invasive spikes (IP) were correctly detected at the edges of spheroids in Fig. 4(l), but the sensitivity (0.346) was low due to mispredicting a few large areas in the ground-truth image [Fig. 4(f)]. The sensitivity was low because the model made many false negative predictions. However, all positive predictions were on the surfaces of the spheroids, which implies that the model has learned that the spikes locate on the edges of the spheroids.

Classes I and IP had very low sensitivity, probably because they shared very similar shapes and textures with other classes. Their training material was not diverse, with a few invasive cells and invasive spikes per image. Many differently aligned subimages were drawn from these examples. Class I was misclassified as P or S and IP as R. They share many similar properties. For example, class I often forms vascular shapes, similar to P, and sharp pointy shapes.

The training material lacked some special cases that occurred in the test images. For example, in image 1 there were two spheroids tightly next to each other with both invasive spikes and cell matter between them. There were no training images for this kind of situation, which could be the reason for the misprediction.

The model trained for class S had quite good scores but struggled with densely populated images. Classes I and R were misclassified as with class S. These classes shared similar properties in size and shape in some cases. In the ground-truth image, some cells around the spheroids were marked as class R and not as noninvasive cells, which in this setting is ambiguous. The cell matter around spheroids is composed of cells. The difference between classes S and R was the distance of the cell from the cluster of cell matter. Taken into account that the markings in the ground-truth images had been made subjectively, the results for S should be taken with a grain of salt in image 1.

Class O would have benefited from more comprehensive data augmentation and optimized threshold selection. As seen in Figs. 5(e) and 5(h), the thresholding has truncated large portions of spheroids where the areas are especially dark. If there would have been enough data for a validation dataset, that data could have been used for selecting a proper threshold using, for example, the high points in the ROCs. More data augmentation with changing image gamma could also have helped in a situation like this.

In terms of the model selection, more exhaustive hyperparameter optimization could have been carried out. The training could have used better weighting for penalizing false positives. Because of the different markings in the data, it was not possible to do class-specific penalization for the background objects. We traded better weighting to a larger number of different training images. However, we could have introduced weighting for all other background classes belonging to a batch. More reliable results would also require comparison with other existing multilabel prediction techniques.

During the experiments, it was noticed that the proliferation rate of the cells greatly influenced how quickly the final growth pattern is reached. Therefore, even though we did not use the primary cells for training the neural network, continuing to image the primary cells for a longer time should be considered, so the final growth pattern would be reached with higher certainty.

The PC3 and LNCaP cells, which were selected for training the neural networks, were not exposed to any drugs, but that will be a goal for future research. Future work would involve using the trained networks to derive data from cancers' invasion patterns. We could also research if the used cell culture model responds to drugs and can be modified to be a personalized cancer model by utilizing patient-derived primary cells in the future.

6 Conclusions

The motivation of this research was to create a personalized medicine model and automate drug efficacy assessment using cancer cell culture microscope images.

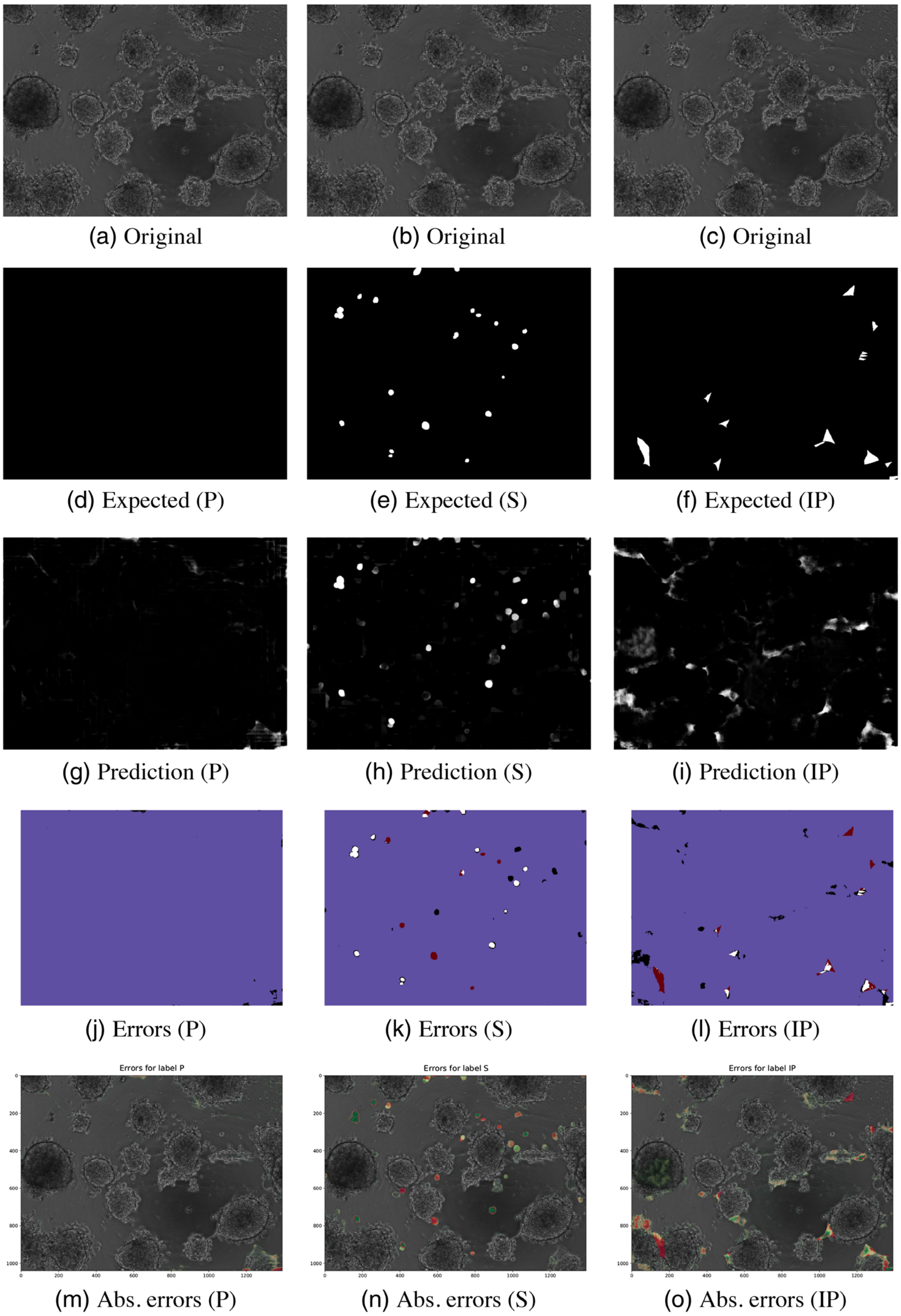
Six structures that were thought to play a part in measuring the drug efficiency were identified. Segmenting captured RGB images was the first step to achieving this. The structures were noninvasively growing cells, invasive cells, invasive spikes, vascular structures, spheroids, and cell matter around the spheroids. The structures could overlap each other.

The dataset consisted of 36 RGB images and their ground-truth images for each class. The data were preprocessed and U-Net neural networks were trained to target each of these classes. The method could distinguish vascular structures, cells, spheroids, and cell matter around spheroids in the test images. Some invasive spikes were also detected, but the method could not distinguish the invasive cells.

The limitations of the study were the lack of data and the imbalanced class distribution, which may question the generalizability of the results. The results suggest that more diverse training data were needed. The results are encouraging, taking the amount of data into account, even though confident conclusions cannot be made. Further research is needed.

Appendix A: Result Images

The predicted images for two test images are illustrated in Figs. 4–7.



(a) Original (b) Original (c) Original

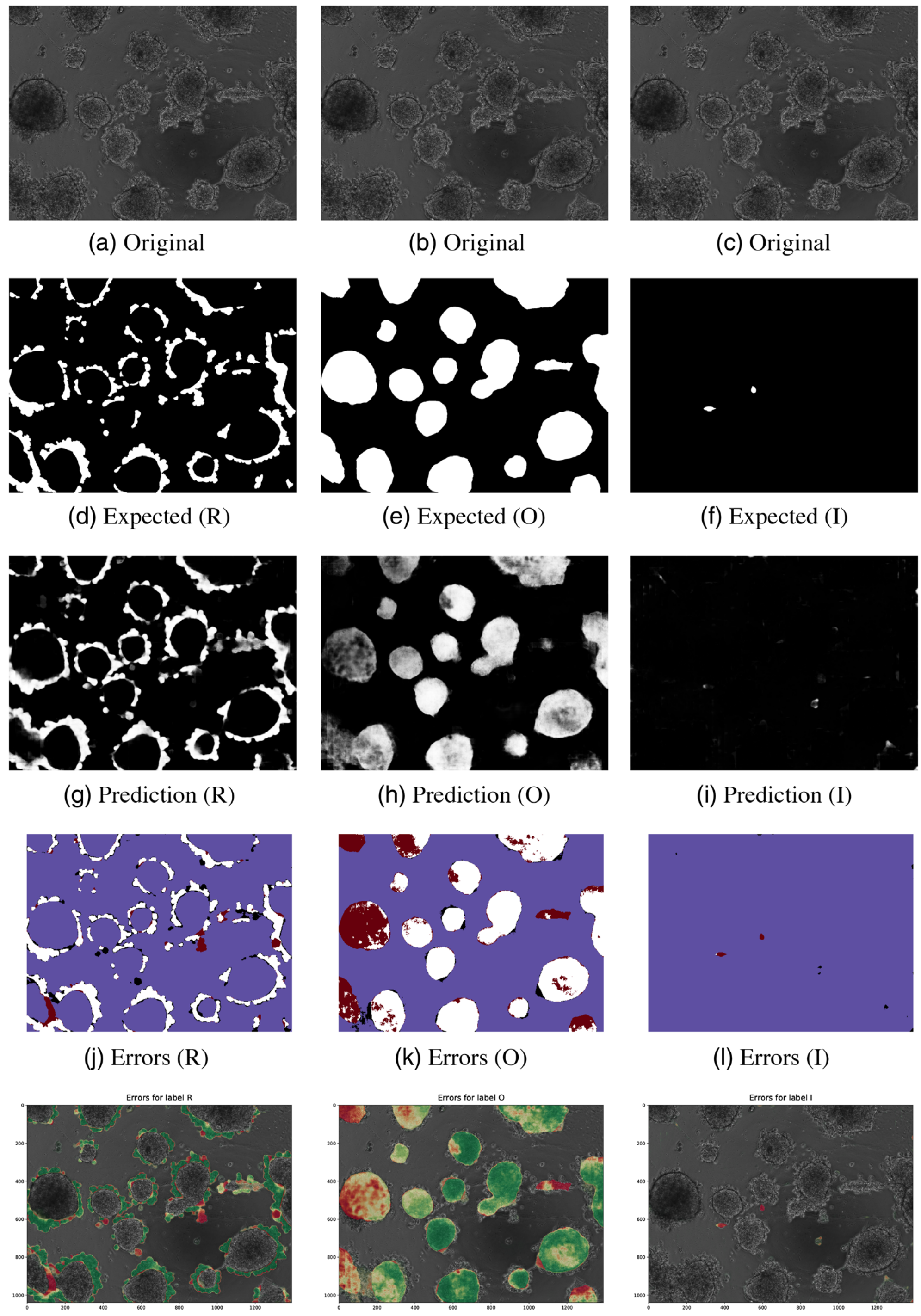
(d) Expected (P) (e) Expected (S) (f) Expected (IP)

(g) Prediction (P) (h) Prediction (S) (i) Prediction (IP)

(j) Errors (P) (k) Errors (S) (l) Errors (IP)

(m) Abs. errors (P) (n) Abs. errors (S) (o) Abs. errors (IP)

Fig. 4 Results for image 1 (PC3) classes P, S, and IP. (a) and (b) The original images, (d)–(f) the ground truths, (g)–(i) the predictions of the models, and (j)–(l) the thresholded pixel classification errors: true positive (white), true negative (blue), false positive (black), and false negative (red). (m)–(o) The absolute errors on the original images (green is more correctly classified and red is wrong).



(a) Original (b) Original (c) Original

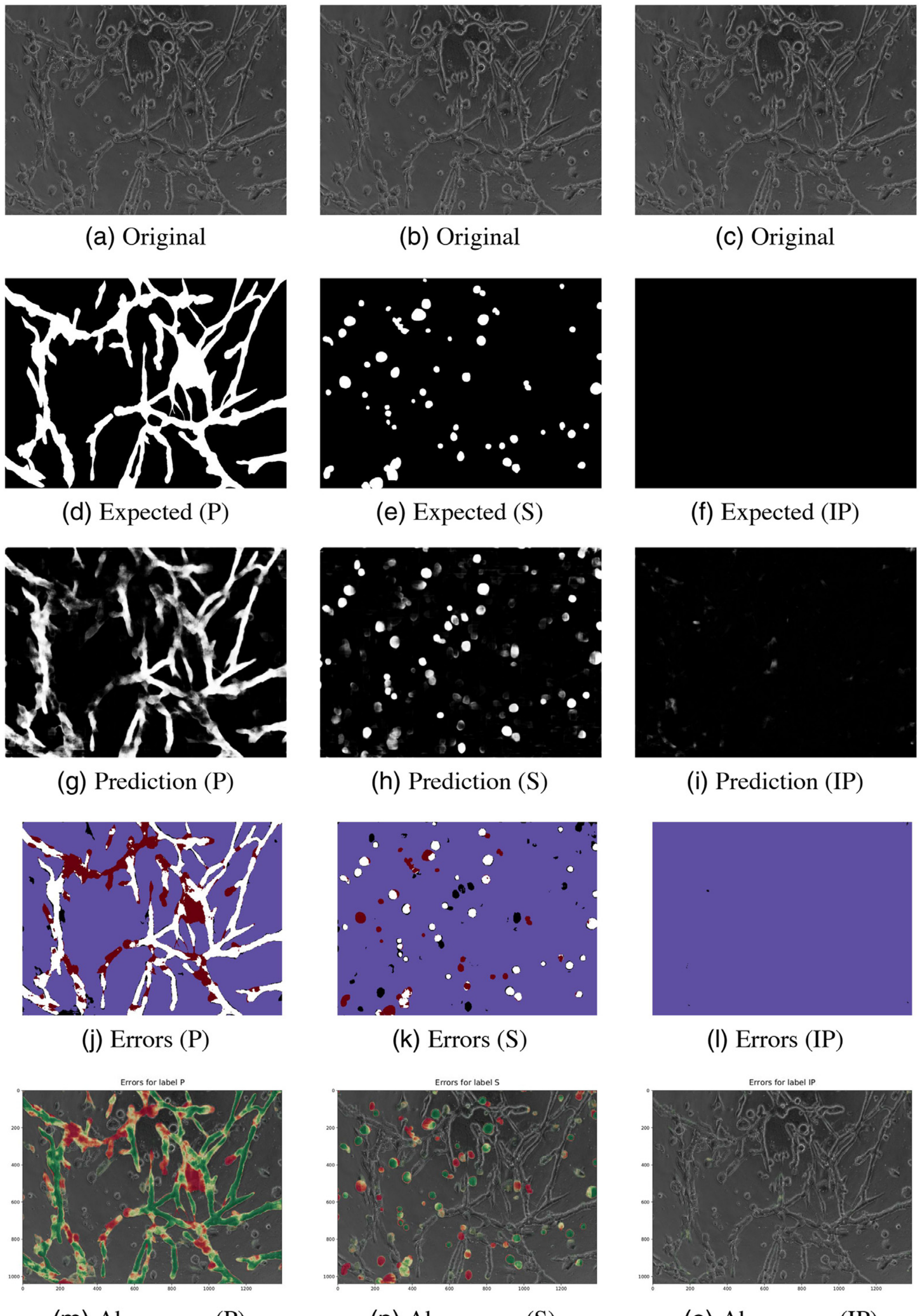
(d) Expected (R) (e) Expected (O) (f) Expected (I)

(g) Prediction (R) (h) Prediction (O) (i) Prediction (I)

(j) Errors (R) (k) Errors (O) (l) Errors (I)

(m) Abs. errors (R) (n) Abs. errors (O) (o) Abs. errors (I)

Fig. 5 Results for image 1 (PC3) classes R, O, and I. (a) and (b) The original images, (d)–(f) the ground truths, (g)–(i) the predictions of the models, and (j)–(l) the thresholded pixel classification errors: true positive (white), true negative (blue), false positive (black), and false negative (red). (m)–(o) The absolute errors on the original images (green is more correctly classified and red is wrong).



(a) Original (b) Original (c) Original

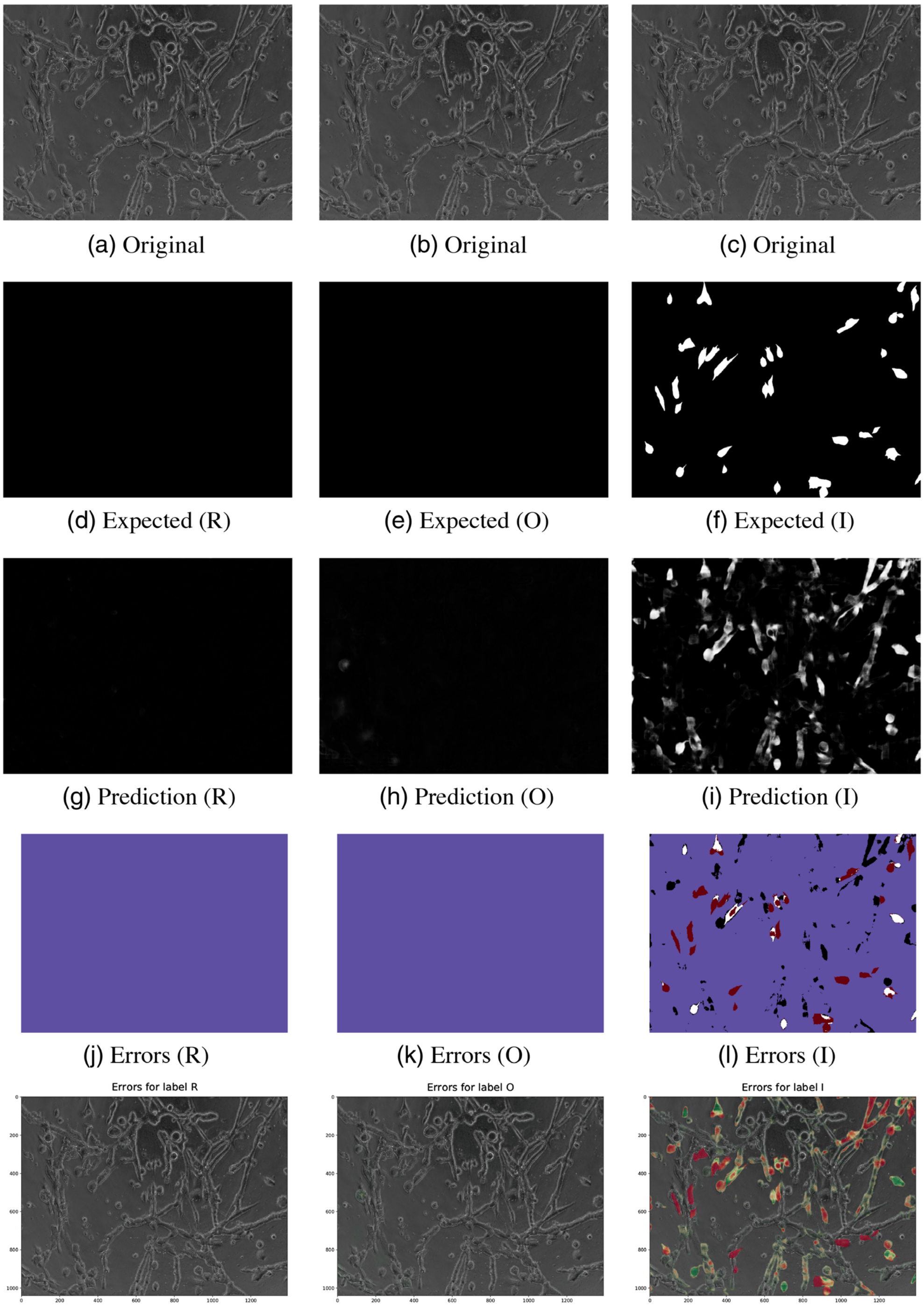
(d) Expected (P) (e) Expected (S) (f) Expected (IP)

(g) Prediction (P) (h) Prediction (S) (i) Prediction (IP)

(j) Errors (P) (k) Errors (S) (l) Errors (IP)

(m) Abs. errors (P) (n) Abs. errors (S) (o) Abs. errors (IP)

Fig. 6 Results for image 2 (LNCaP) classes P, S, and IP. (a) and (b)The original images, (d)–(f) the ground truths, (g)–(i) the predictions of the models, and (j)–(l) the thresholded pixel classification errors: true positive (white), true negative (blue), false positive (black), and false negative (red). (m)–(o) The absolute errors on the original images (green is more correctly classified and red is wrong).



(a) Original (b) Original (c) Original

(d) Expected (R) (e) Expected (O) (f) Expected (I)

(g) Prediction (R) (h) Prediction (O) (i) Prediction (I)

(j) Errors (R) (k) Errors (O) (l) Errors (I)

(m) Abs. errors (R) (n) Abs. errors (O) (o) Abs. errors (I)

Fig. 7 Results for image 2 (LNCaP) classes R, O, and I. (a) and (b) The original images, (d)–(f) the ground truths, (g)–(i) the predictions of the models, and (j)–(l) the thresholded pixel classification errors: true positive (white), true negative (blue), false positive (black), and false negative (red). (m)–(o) The absolute errors on the original images (green is more correctly classified and red is wrong).

Disclosures

The authors have no relevant financial interests in the article and no other potential conflicts of interest to disclose.

Acknowledgments

We would like to thank Outi Huttala (PhD in cell biology) for valuable help with the cell cultures. The research has been cofunded by University of Jyväskylä, the Finnish Funding Agency for Innovation Tekes (Grant No. 1711/31/2016) and the Foundation of Jane and Aatos Erkko (Grant No. 170015).

Use of Humans and Animals

This study was conducted in the spirit of Good Laboratory Practice Regulations as set forth in OECD [ENV/MC/CHEM(98)17] and according to the relevant Standard Operating Procedures of FICAM. Humans or animals were not used in this study.

The study conforms to the ethical principles outlined in the Declaration of Helsinki. The human adipose tissue samples were obtained from the excess material of surgical operations. Human umbilical cords were received from caesarean sections with written informed consents at Tampere University Hospital (Tampere, Finland). The use of human adipose stromal cells and human umbilical cord endothelial cells were approved by the Ethics Committee of the Pirkanmaa Hospital District (Tampere, Finland) with the permit numbers R15161 and R15033.

References

1. C. Davis et al., "Availability of evidence of benefits on overall survival and quality of life of cancer drugs approved by European Medicines Agency: retrospective cohort study of drug approvals 2009–13," *BMJ* **359**, j4530 (2017).
2. V. Prasad, K. De Jesus-Morales, and S. Mailankody, "The high price of anticancer drugs: origins, implications, barriers, solutions," *Nat. Rev. Clin. Oncol.* **14**, 381–390 (2017).
3. R. Edmondson et al., "Three-dimensional cell culture systems and their applications in drug discovery and cell-based biosensors," *ASSAY Drug Dev. Technol.* **12**, 207–218 (2014).
4. V. Härmä et al., "A comprehensive panel of three-dimensional models for studies of prostate cancer growth, invasion and drug responses," *PLoS One* **5**, e10431 (2010).
5. P. Kenny et al., "The morphologies of breast cancer cell lines in three-dimensional assays correlate with their profiles of gene expression," *Mol. Oncol.* **1**, 84–96 (2007).
6. V. Liarski et al., "Quantifying in situ adaptive immune cell cognate interactions in humans," *Nat. Immunol.* **20**, 503–513 (2019).
7. M. Kress et al., "Time-resolved microspectrofluorometry and fluorescence lifetime imaging of photosensitizers using picosecond pulsed diode lasers in laser scanning microscopes," *J. Biomed. Opt.* **8**(1), 26–32 (2003).
8. V. M. Liarski et al., "Cell distance mapping identifies functional T follicular helper cells in inflamed human renal tissue," *Sci. Transl. Med.* **6**(230), 230ra46 (2014).
9. I. Ahonen et al., "A high-content image analysis approach for quantitative measurements of chemosensitivity in patient-derived tumor microtissues," *Sci. Rep.* **7**(1), 6600 (2017).
10. O. Ronneberger, P. Fischer, and T. Brox, "U-Net: convolutional networks for biomedical image segmentation," *Lect. Notes Comput. Sci.*, N. Navab et al., Eds., vol. **9351**, 234–241, Springer, Cham, Germany (2015).
11. A. A. Novikov et al., "Fully convolutional architectures for multiclass segmentation in chest radiographs," *IEEE Trans. Med. Imaging* **37**(8), 1865–1876 (2018).
12. E. M. Christiansen et al., "*In silico* labeling: predicting fluorescent labels in unlabeled images," *Cell* **173**(3), 792-803.e19 (2018).
13. S. Nikolov et al., "Deep learning to achieve clinically applicable segmentation of head and neck anatomy for radiotherapy," arXiv: abs/1809.04430 (2018).
14. L.-C. Chen et al., "Encoder–decoder with atrous separable convolution for semantic image segmentation," *Lect. Notes Comput. Sci.* **11211**, 833–851 (2018).
15. P. Isola et al., "Image-to-image translation with conditional adversarial networks," in *IEEE Conf. Comput. Vision and Pattern Recognit. (CVPR)*, Honolulu, Hawaii, pp. 5967–5976 (2017).
16. X. Chen et al., "Focus, segment and erase: an efficient network for multi-label brain tumor segmentation," *Lect. Notes Comput. Sci.* **11217**, 674–689 (2018).
17. A. Maxwell et al., "Deep learning architectures for multi-label classification of intelligent health risk prediction," *BMC Bioinf.* **18**, 523 (2017).

18. O. Huttala et al., "Human vascular model with defined stimulation medium—a characterization study," *ALTEX* **32**, 125–136 (2015).
19. W. H. Ng et al., "Extracellular matrix from decellularized mesenchymal stem cells improves cardiac gene expressions and oxidative resistance in cardiac c-kit cells," *Regener. Ther.* **11**, 8–16 (2019).
20. M. Abadi, "TensorFlow: large-scale machine learning on heterogeneous systems," (2015).
21. K. Tarek, "GitHub repository 'U-Net'," https://github.com/kimoktm/U-Net (2017).
22. D. Kingma and J. Ba, "Adam: a method for stochastic optimization," in *Int. Conf. Learn. Represent.* (2014).
23. X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. Thirteenth Int. Conf. Artif. Intell. and Stat.*, Y. W. Teh and M. Titterington, Eds., PMLR, Sardinia, Italy, Vol. 9, pp. 249–256 (2010).
24. K. H. Zou et al., "Statistical validation of image segmentation quality based on a spatial overlap index," *Acad. Radiol.* **11**(2), 178–189 (2004).
25. P. Sonego, A. Kocsor, and S. Pongor, "ROC analysis: applications to the classification of biological sequences and 3D structures," *Briefings Bioinf.* **9**(3), 198–209 (2008).

Samuli Rahkonen is a PhD student at the University of Jyväskylä and works in the Spectral Imaging Laboratory as a project researcher and in the private sector as a software engineer. In 2015, Rahkonen finished his MSc (Tech) degree in computer and software engineering at Tampere University of Technology. His research interests lie in applied machine learning methods. These include using cell culture microscopy imaging and spectral data with neural networks in medical pattern recognition.

Emilia Koskinen graduated from the University of Turku with a bachelor's degree in biomedicine and a master's degree in drug discovery and development in 2016. She is interested in human health, toxicology, and *in-vitro* model system development.

Ilkka Pölönen is the head of Spectral Imaging Laboratory at the Faculty of Information Technology, University of Jyväskylä. His research interests are in the fields of spectral imaging, machine vision, data analysis, mathematical modeling, and numerical simulations.

Tuula Heinonen is a European registered toxicologist having deep theoretical education and over 25 years' experience in toxicology in industry and academia. She has been responsible for setting up the Finnish Centre for Alternative Methods (FICAM) and is its director. FICAM develops validated tissue and organ models to supplement and replace animal experiments, educate scientists, and share information. Her publications cover toxic risk assessment of chemicals, testing strategies, and development and validation of new *in-vitro* tests.

Timo Ylikomi (MD, PhD) is the head of Department of Cell Biology in the Faculty of Medicine and Health Technology at the University of Tampere. He is the author of over 100 peer-reviewed papers published in international journals. As a medical doctor by education, he has much experience in cell biology, cell culture, and human cell-based engineered tissues and cancer biology. He started studying adipose-derived stem cells in the early 2000s with a goal of developing bioengineered tissue products for human use and for testing purposes. One important research interest of his is bioengineered soft tissue based on bioactive angiogenic and adipogenic substances.

Sami Äyrämö is an adjunct professor of data analytics at the University of Jyväskylä. He received his PhD in mathematical information technology in 2006. He also has an MSc degree in sports sciences. His research interests include machine learning and predictive modeling with applications in sport, health, and medicine. He is also leading a group of researchers with a special focus on machine learning in health and medicine.

Matti A. Eskelinen is a PhD student at the Spectral Imaging Laboratory of the Faculty of Information Technology at the University of Jyväskylä. He received his MSc degree in theoretical physics from the University of Jyväskylä in 2015, and has since been studying computational methods in hyperspectral imaging. His research interests include hyperspectral image analysis and data processing, machine learning, and inverse problems.

PIV

METHOD FOR RADIANCE APPROXIMATION OF HYPERSPECTRAL DATA USING DEEP NEURAL NETWORK

by

Samuli Rahkonen and Ilkka Pölönen 2023

Impact of scientific computing on science and society, Springer, pending publication

Method for Radiance Approximation of Hyperspectral Data Using Deep Neural Network

Samuli Rahkonen and Ilkka Pölönen

Abstract We propose a neural network model for calculating the radiance from raw hyperspectral data gathered using a Fabry–Perot interferometer color camera developed by VTT Technical Research Centre of Finland. The hyperspectral camera works by taking multiple images from different wavelength with varying interferometer settings. The raw data needs to be converted to radiance in order to make any use of it, but this leads to larger file sizes. Because of the amount of the data and the structure of the raw data, the processing has to be run in parallel, requiring a lot of memory and time. Using raw camera data could save processing time and file space in applications with computation time requirements. Secondly, this kind of neural network could be used for generating synthetic training data or use it in generative models. The proposed model approaches these problems by combining spatial and spectral-wise convolutions in neural network with minimizing a loss function utilizing the spectral distance and mean squared loss. The used dataset included images from many patients with melanoma skin cancer.

1 Introduction

VTT Technical Research Centre of Finland has developed a prototype hyperspectral camera based on a Fabry–Perot interferometer color camera. The camera works by taking multiple images from different wavelength with varying interferometer settings resulting to a set of raw sensor RGB images. Processing the large data cubes produced by these cameras is time and memory consuming. Traditionally the data must have been converted to radiance data in order to make any use of it in applications, but this leads to large file sizes. The conversion step could be skipped completely by using the raw data, for example, in target segmentation in medical

Samuli Rahkonen (✉) · Ilkka Pölönen
University of Jyväskylä, Faculty of Information Technology, P.O. Box 35, FI-40014 University of Jyväskylä, Finland, e-mail: samuli.rahkonen@jyu.fi

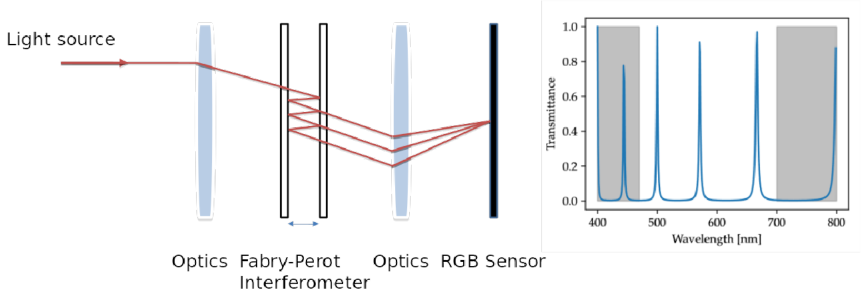


Fig. 1 Working principle of Fabry–Perot interferometer camera

imaging. Secondly, this kind of neural network could be used for generating synthetic training data or use it in generative models, like generative adversial networks.

We propose a neural network model for calculating the radiance from the raw data produced by this kind of hyperspectral cameras. Figure 1 illustrates how a Fabry–Perot interferometer works. It has two parallel half-mirrors close to each other. A beam of light entering the system interferes with itself as it reflects off the mirrors. Integer multiples of light with certain wavelengths are then transmitted through the mirrors. With low- and high-pass filters, the setup can be used as a narrowband wavelength filter by controlling the mirror separation. [2, 7]

Conversion to radiance data cube is traditionally done as follows. The raw RGB sensor data is bilinearly interpolated (using a Bayer matrix) to produce an array of RGB images. After interpolation, each image can be converted to multiple narrow wavelength radiance bands depending on the number of received peaks on the sensor. Needed coefficients for calculating the radiance for corresponding wavelengths are solved during the camera calibration. Therefore, the number of bands in the resulting radiance cube can be higher than in the original raw data cube. Finally, the produced bands are sorted by their wavelength to produce the final radiance cube. [2, 7]

This paper introduces a neural network architecture for predicting a radiance cube from a given raw sensor data cube. We demonstrate the network's ability to calculate the radiance using a dataset of images of skin melanoma. We noticed reduced noise in the produced radiance cubes.

2 Methods

The experiments were ran on a shared computer cluster with Intel(R) Xeon(R) CPU E5-2640 v4, 264 GB memory and Nvidia Tesla P100 GPU with 16GB memory. We used Python 3.6 and Tensorflow 2.0.0.

2.1 Dataset and Network Architecture

Our dataset consisted of 62 raw hyperspectral images captured with VTT Fabry–Perot interferometer camera with 1920×1080 spatial resolution and 85 bands. A raw image is not yet converted to radiance image and its bands have not been interpolated using the RGB Bayer matrix on the photo sensor. The images were captured from patients with and without diagnosed melanoma cancer. Figure 2 shows three example images from our dataset. We assume that a general method for radiance approximation using a neural network would require a very large and diverse dataset of hyperspectral data, which was not available for this type of camera. Therefore we are using this dataset and narrowing our research scope to be application specific to hyperspectral images of skin.

In order to create a ground truth dataset, we used `fpipy` Python library [5] to generate radiance images from the same raw images. The resulting radiance cubes have 120 bands. The ground truth raw and radiance data included noise. The dataset was split to 54 training and 8 test image cubes by hand to make sure we had diverse sets of images available. Data pipeline normalized the full image cubes between 0 and 1, extracted 21 subimages ($256 \times 256 \times 85$ and $256 \times 256 \times 120$) patches from each raw and its corresponding radiance data cubes. During the training, images were also rotated and mirrored in spatial domain to enlarge the data pool.

The network consists of 3D convolution layers with filters applied at different directions inside the data cube (see Fig. 3). First, we expand dimensions by one (at the end of the tensor), to apply the 3D convolution filters inside spatial and spectral axis of the cube. Also the 3D convolution layer expects 5D tensors as an input. The two $3 \times 3 \times 1$ kernels are used at spatial plane for each band to find filters for bilinear interpolation. We use batch normalization and LeakyRelu [6] as the activation function of the hidden units to avoid gradient vanishing/exploding problem. The number of filters and their shape are kept small, as traditionally the bilinear interpolation of a raw Bayer matrix image could be achieved with convolving just two 3×3 kernels.

Next, we use two consecutive layers with $1 \times 1 \times 5$ kernels at spectral axis. The idea is to only look at the spectra of the pixels to find filters with suitable coefficients which can convert the out-of-order raw sensor signals to radiance. They are out of order

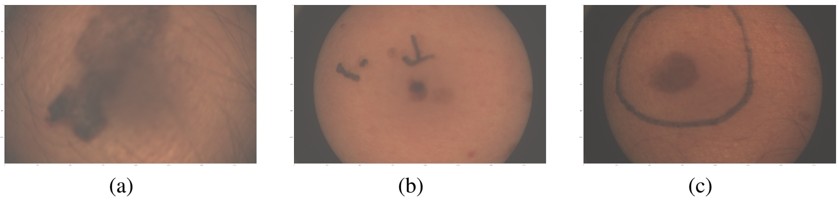

(a) (b) (c)

Fig. 2 Examples of hyperspectral images (including only RGB bands for visualization) from patients; handmade markings on the skin in images (b) and (c)

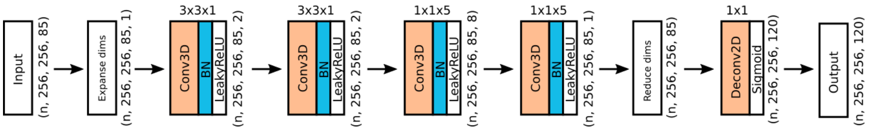


Fig. 3 Neural network model

because each band in the data cube can converted to multiple narrow wave-length radiance bands depending on the number of received peaks on the sensor.

We drop the last dimension to upscale the spectral dimension. At the end of the network, we use deconvolution layer with sigmoid activation function to increase the number of bands from 85 to 120, which is the expected number of bands. Sigmoid function outputs activation values between 0 and 1.

2.2 Training

We trained the network with 205 steps. Each step included 100 batches of two training input-output data pairs. The batch size was 2. Training with full epochs of all training images slowed down the updating of the weights too much. The GPU memory and cube size were the limiting factors. Too long training would result to the network to start modeling the noise in the training data. This could be seen from individual testing spectra during the training.

The loss function is

$$\begin{aligned}\text{Loss} = \lambda_1 \frac{1}{NMB} \sum_i^N \sum_j^M \sum_k^B (\boldsymbol{Y}_{ijk} - \hat{\boldsymbol{Y}}_{ijk})^2 \\ + \lambda_2 \frac{1}{NM} \sum_i^N \sum_j^M \theta(\boldsymbol{Y}_{ij}, \hat{\boldsymbol{Y}}_{ij}) + \|\boldsymbol{W}\|_2, \end{aligned} \tag{1}$$

$$\theta(\boldsymbol{S}, \hat{\boldsymbol{S}}) = \cos(\boldsymbol{S}, \hat{\boldsymbol{S}}) = \frac{\boldsymbol{S} \cdot \hat{\boldsymbol{S}}}{\|\boldsymbol{S}\|_2 \|\hat{\boldsymbol{S}}\|_2}, \tag{2}$$

where $\boldsymbol{Y}$ and $\hat{\boldsymbol{Y}}$ are the ground truth and predicted training data cubes, respectively. Pixel counts N and M are at the spatial axis and B is the number of bands at the spectral axis.

The first term is the mean square error for minimizing the spatial errors and errors in pixel intensity values. The experimentally defined coefficient $\lambda_1 = 1$ gives a weight for this term. The second term is the mean of spectral angles between predicted and expected spectra ($\boldsymbol{S}$ and $\hat{\boldsymbol{S}}$) in a pixel. It uses Eq. (2) for calculating the spectral angle. This is used to minimize the errors at the spectral axis and keep the form of the spectra closer to the ground truth data. The effect of this term to the predicted spectra was qualitatively validated during experiments. The experimentally defined coefficient

Table 1 Test results

Cube number	MAE	MSE	PSNR [dB]	SSIM
1	0.060	0.005	22.877	0.574
2	0.052	0.004	23.933	0.813
3	0.078	0.009	20.457	0.809
4	0.084	0.008	20.468	0.565
5	0.047	0.002	25.354	0.725
6*	0.040	0.002	26.073	0.831
7	0.029	0.001	29.208	0.831
8	0.038	0.002	26.405	0.831

* Results described and discussed in detail

$\lambda_2 = 10$ gives a weight for this term. The third term is the L2 regularization for the network weights to prevent overfitting and to keep the training more stable by avoiding gradient explosion. We used Adam optimizer with a learning rate of $2e^{-4}$ and $\varepsilon = 0.01$. The gradients were set to be clipped by norm 1 for stability. Training the neural network took three hours with the GPU.

We compared the radiance cube generated by the network to the radiance calculated by `fpipy` library [5]. `fpipy` calculates the radiance (see Sect. 1). We used several quality metrics and qualitative visual inspection to measure differences between these two radiance calculation methods. The metrics included structural similarity index measure (SSIM) [10, 11], peak signal to noise ratio (PSNR), mean absolute error (MAE) and mean squared error (MSE).

SSIM can be used for measuring the perceived image quality and ranges between -1 and 1, where value 1 would mean that the input images are equal. SSIM takes better into account the structural information (texture, orderings, patterns) in natural signals and should provide better metric for the experimented dataset [10]. Scikit-image [9] implementation of SSIM was used.

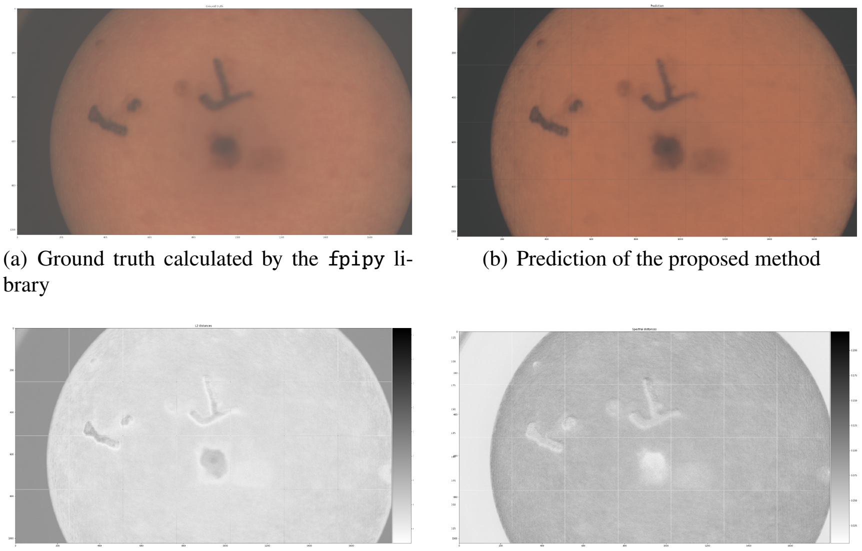
3 Results

To interpret the MAE and MSE, we consider possible cube values between 0 and 1. The ranges of the metrics were as follows:

- MAE: 0.02–0.09,
- MSE: 0.001–0.009,
- PSNR: 20–30 dB,
- SSIM: 0.5–0.9.

Predicting a full image cube took 4.2–6.8 seconds with the neural network and the GPU. Radiance calculation with `fpipy` took approximately 48 seconds with the CPU. The results are listed in Table 1.

The cube number 6 was selected as a representative example. Different plots illustrating the differences between the ground truth and the predicted cube are shown



(a) Ground truth calculated by the `fpipy` library

(b) Prediction of the proposed method

(c) Pixel-wise L2 error between the ground truth and the prediction

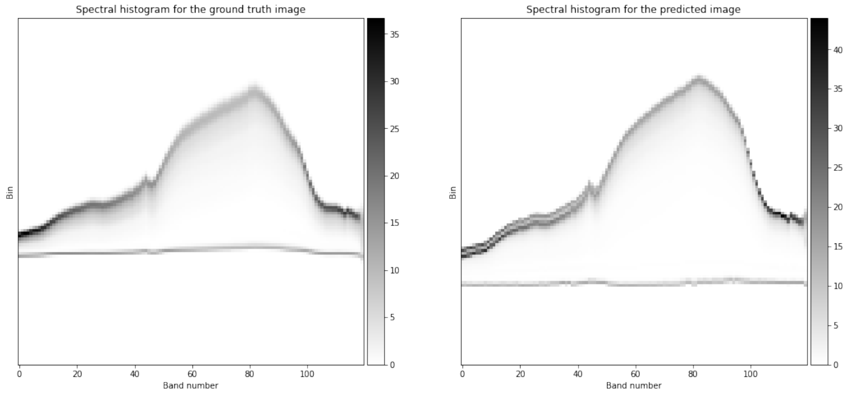
(d) Pixel-wise spectral distance (2) between the ground truth and the prediction

Fig. 4 Visualization of the cube number 6 with the RGB bands

in Figs. 4–7. Figure 4 illustrates errors between the two methods. The prediction has visible cut lines between the image patches, because the network tends to create artifacts at the edges of the image that are visible after the full image cube is reconstructed.

Comparing the ground truth (Fig. 5(a)) and prediction (Fig. 5(b)) histograms, you can see predicted pixels between bands 69–90 to gain larger values compared to the ground truth. These bands correspond to red wavelengths. The errors can be seen in the histograms, in Fig. 6 around the corresponding wavelengths 625–700, and also in the RGB image (Fig. 4) as a reddish tint.

Based on visual inspection and Fig. 7, which illustrates one image patch and two example spectra, the network seems to reduce noise in the example spectra. Also, Fig. 5(a) shows that the ground truth bands have more variance in their data values than in the prediction (Fig. 5(b)). The last two bands have a lot of noise and cause large error peaks. The prediction image showed some noticeable color alterations in recurring pattern at spatial dimensions at a closer visual inspection. This is probably due to the fact that the network fails to completely reproduce the bilinear interpolation for the Bayer color filter array of the camera's photosensor.



(a) Ground truth

(b) Predicted

Fig. 5 Histograms of the image spectra; 120 bands on the X-axis, 120 bins for each band on the Y-axis

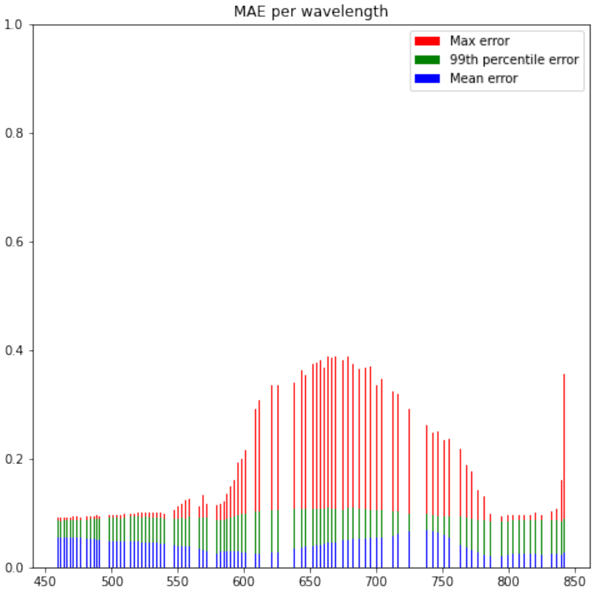


Fig. 6 MAE per wavelength for the cube number 6; wavelength on the X-axis, absolute error on the Y-axis

4 Discussion

In this section, we discuss the main findings and the related work on the topic.

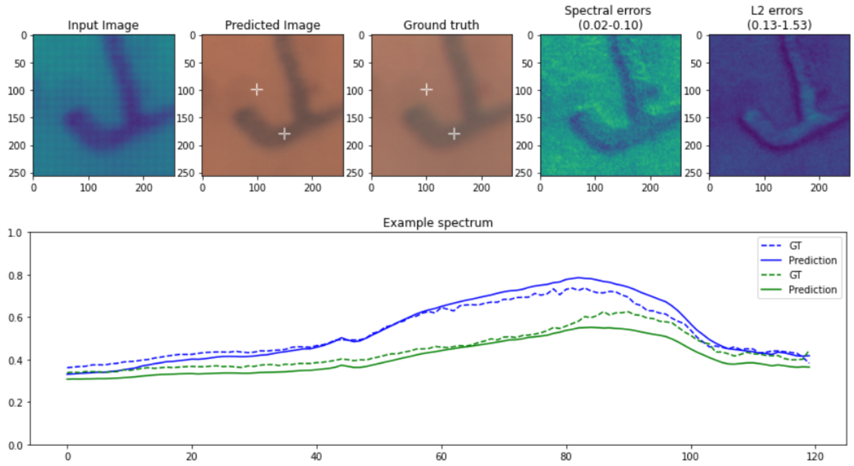


Fig. 7 Top: example spectra and errors of a test cube patch in the cube number 6; Bottom: spectrum from the upper point (blue) and the lower point (green) in the patch

4.1 Findings

The PSNR was between 20–30 dB and the MAE between 0.02–0.09, which are quite good results. The largest MAE errors are in the range of 670–775 nm, as shown in Fig. 6. These bands had the most variance in pixel values, which explains the errors. The SSIM varied between 0.57–0.83, which can be interpreted as a good result. The ground truths of the cubes number 1 and 4 were quite noisy at spectral dimension and unfocused spatially, but their spectral histograms for the predicted cubes show much less variance per band. As a result, they scored worse than the other test cubes (see Table 1). Visual inspection to the zoomed in ground truth, prediction and error images in Fig. 7 show that the results are quite close to each other at both spatial and spectral dimensions. The largest L2 spectral errors are at the areas with least samples of similar spectra.

The prediction histogram (Fig. 5(b)) illustrates how the spectra of the proposed method follows quite closely the ground truth method. At a closer look, the ground truth histogram shows some repeating error patterns and larger variance of the spectra values throughout the bands. The intensity values of the predicted spectra are more tightly concentrated. This could be a result of overgeneralizing the training data, or the network is reducing the noise.

A strange effect is observed from the prediction histogram (Fig. 5(b)). Bands between 0–50 form three band intensity peaks instead of two as in the ground truth (Fig. 5(a)). This can be explained by the used cube patching method. The full cube was split into patches which were fed through the neural network and the predictions were assembled back to a larger cube. The method could predict each patch differently, which could show in the histogram like this.

4.2 Related Work

There exist many cases where neural networks are used for image processing, where the mapping from input to output images is learnt from training images. For Bayer filter interpolation there are a few attempts on applying neural networks [4, 8]. Convolutional neural networks has been used for demosaicing photosensor image data [8]. The authors quantitatively and qualitatively compare ten demosaicing algorithms to two methods (DMCNN and DMCNN-VD) based on neural networks. DMCNN uses a few layers with the intent to learn features for interpolation, non-linearly mapping them to individual pixels and reconstructing them to a full image. The second method, DMCNN-VD, uses residual layers to construct a much deeper network. The results look promising, but artifacts are still present.

In the case of spatial interpolation, gathering and constructing a good ground truth dataset is a challenge. One way to enhance the existing data is to create a pseudo ground truth dataset by spatially downsampling the training images, adding noise and creating mosaic from them to reduce noise [4]. Generative adversial networks (GAN) have been used with hyperspectral data for visualization [1], image restoration [3] and generating new images. They have even been used for learning mappings between unpaired RGB images [12].

5 Conclusions

The aim of the research was to find a method using a neural network for approximating the radiance from the raw data produced by a FPI camera. The proposed neural network method delivers promising results, but suffers from some image artifacts in the form of per band intensity fluctuations depending on the quality of the raw input data. The model is also able to reduce noise of the spectra with noisy training data.

As topics for future research, we would consider producing better training data. Bilinear interpolation can be achieved quite fast without a neural network and therefore it would make sense to do this step as a separate preprocessing operation before running a radiance approximation. Alternatively we could have undersampled the image data spatially to produce less noisy ground truth data for training.

We noticed that the training was not completely stable, because of fluctuations in image sharpness between training steps in spatial dimensions. At spectral dimension, we noticed varying which caused intensities of all the band values to be off. The reason is probably the loss function which emphasizes the shape of the spectra over the intensity. One solution could be optimizing the λ_1 and λ_2 coefficients in the loss function. Secondly, the learning rate could be decreased during the training.

We experimented on using just MAE as a loss function for the network, but it failed to capture the spatial and spectral fidelity of the ground truth data. The predictions tended to capture and produce only the mean spectra of the whole image. Using spectral angle (2) in the loss function made it possible to distinguish different spectra and MAE contributed to the spatial image quality. The results show that it is possible to train an application specific neural network for radiance approximation with a relatively small number of training images.

Acknowledgements This research was supported by the University of Jyväskylä.

References

1. S. Chen, D. Liao, and Y. Qian. Spectral image visualization using generative adversarial networks. In X. Geng and B.H. Kang, editors, *PRICAI 2018: Trends in Artificial Intelligence*, volume 11012 of *Lecture Notes in Computer Science*, pages 388–401, Cham, 2018. Springer.
2. M. Eskelinen. *Computational methods for hyperspectral imaging using Fabry–Perot interferometers and colour cameras*. PhD thesis, University of Jyväskylä, 2019.
3. C. Fabbri, M.J. Islam, and J. Sattar. Enhancing underwater imagery using generative adversarial networks. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7159–7165. IEEE, 2018.
4. M. Gharbi, G. Chaurasia, S. Paris, and F. Durand. Deep joint demosaicking and denoising. *ACM Trans. Graph.*, 35(6), 2016.
5. J. Hämäläinen, S. Jääskeläinen, and S. Rahkonen. Fabry-Perot imaging in Python. GitHub, https://github.com/silmae/fpipy, 2018.
6. A.L. Maas, A.Y. Hannun, and A.Y. Ng. Rectifier nonlinearities improve neural network acoustic models. In *ICML'13: Proceedings of the 30th International Conference on International Conference on Machine Learning*, volume 28 of *JMLR Workshop and Conference Proceedings*. JMLR.org, 2013.
7. H. Saari, V.-V. Aallos, A. Akujärvi, T. Antila, C. Holmlund, U. Kantojärvi, J. Mäkynen, and J. Ollila. Novel miniaturized hyperspectral sensor for UAV and space applications. In R. Meynart, editor, *Sensors, Systems, and Next-Generation Satellites XIII*, volume 7474 of *Proceedings of SPIE*. International Society for Optics and Photonics SPIE, 2009.
8. N.S. Syu, Y.S. Chen, and Y.Y. Chuang. Learning deep convolutional networks for demosaicing. arXiv:1802.03769, 2018.
9. S. van der Walt, J.L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J.D. Warner, N. Yager, E. Gouillart, T. Yu, and the scikit-image contributors. scikit-image: image processing in Python. *PeerJ*, 2:e453, 2014.
10. Z. Wang and A.C. Bovik. Mean squared error: Love it or leave it? a new look at signal fidelity measures. *IEEE Signal Processing Magazine*, 26(1):98–117, 2009.
11. Z. Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.
12. J. Zhu, T. Park, P. Isola, and A.A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2242–2251. IEEE, 2017.

PV

REFLECTANCE MEASUREMENT METHOD BASED ON SENSOR FUSION OF FRAME-BASED HYPERSPECTRAL IMAGER AND TIME-OF-FLIGHT DEPTH CAMERA

by

Samuli Rahkonen, Leevi Lind, Anna-Maria Raita-Hakola, Sampsa Kiiskinen and Ilkka Pölönen 2022

MDPI Sensors, Sensors and Robotics, Kinect Sensor and Its Application

sensors

MDPI

Article

Reflectance Measurement Method Based on Sensor Fusion of Frame-Based Hyperspectral Imager and Time-of-Flight Depth Camera

Samuli Rahkonen *, Leevi Lind, Anna-Maria Raita-Hakola, Sampsa Kiiskinen and Ilkka Pölönen

Faculty of Information Technology, University of Jyväskylä, 40014 Jyväskylä, Finland

* Correspondence: samuli.rahkonen@jyu.fi

Abstract: Hyperspectral imaging and distance data have previously been used in aerial, forestry, agricultural, and medical imaging applications. Extracting meaningful information from a combination of different imaging modalities is difficult, as the image sensor fusion requires knowing the optical properties of the sensors, selecting the right optics and finding the sensors' mutual reference frame through calibration. In this research we demonstrate a method for fusing data from Fabry–Perot interferometer hyperspectral camera and a Kinect V2 time-of-flight depth sensing camera. We created an experimental application to demonstrate utilizing the depth augmented hyperspectral data to measure emission angle dependent reflectance from a multi-view inferred point cloud. We determined the intrinsic and extrinsic camera parameters through calibration, used global and local registration algorithms to combine point clouds from different viewpoints, created a dense point cloud and determined the angle dependent reflectances from it. The method could successfully combine the 3D point cloud data and hyperspectral data from different viewpoints of a reference colorchecker board. The point cloud registrations gained 0.29–0.36 fitness for inlier point correspondences and RMSE was approx. 2, which refers a quite reliable registration result. The RMSE of the measured reflectances between the front view and side views of the targets varied between 0.01 and 0.05 on average and the spectral angle between 1.5 and 3.2 degrees. The results suggest that changing emission angle has very small effect on the surface reflectance intensity and spectrum shapes, which was expected with the used colorchecker.

Keywords: hyperspectral; depth data; kinect; sensor fusion; reflectance

check for updates

Citation: Rahkonen, S.; Lind, L.; Raita-Hakola, A.-M.; Kiiskinen, S.; Pölönen, I. Reflectance Measurement Method Based on Sensor Fusion of Frame-Based Hyperspectral Imager and Time-of-Flight Depth Camera. *Sensors* **2022**, *22*, 8668. https://doi.org/10.3390/s22228668

Academic Editor: Seung-Chul Yoon

Received: 29 September 2022
Accepted: 8 November 2022
Published: 10 November 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Extracting meaningful information from a combination of different imaging modalities, such as standard RGB images, hyperspectral data and depth maps produced by depth perceiving cameras, is a demanding task. Fusing data of different types, volumes and dimensions from varying sources and different sensors is a research area with a lot of emerging new technologies and applications.

Image sensor fusion requires knowing the optical properties of the sensors, selecting the right optics and the finding sensors' mutual reference frame through calibration. In our case, producing a hyperspectral point cloud also requires estimating the relative positions and orientations of the cameras in the world by using registration algorithms.

Hyperspectral imaging (HSI) considers capturing images with specialized hyperspectral cameras. Each image pixel captures a spectrum of light and each wavelength is captured with a narrow bandwidth. The spectral and spatial dimensions together can be used to characterize and identify points of interest in the image [1].

Previously, depth sensing imaging technologies have shown their ability to add meaningful information to improve, e.g., classification [2], robot navigation [3], and segmenting regions of interest from images [3,4]. In this research, we are using Kinect V2 depth camera.

Depth sensing cameras are able to capture depth maps where each pixel corresponds to a distance.

We used Piezo-actuated metallic mirror Fabry–Pérot interferometer (FPI) hyperspectral camera, which is a frame-based imager, developed at the Technical Research Centre of Finland Ltd (VTT) [5]. It captures the scene by taking multiple frames and combining them into a hyperspectral data cube with spatial and spectral dimensions. Each produced pixel in the image corresponds to a mixed radiance spectrum of light, ranging from visible light to infrared wavelengths, depending on the application and the camera. A common quantity measured with these devices is the spectral reflectance of a material, defined as the ratio of reflected and incident light per measured wavelength band.

Frame-based hyperspectral cameras produce an image from a static target without moving the camera itself, as opposed to the whisk broom or push broom type of scanners [6]. Using a frame-based imager makes it easier to fuse the sensor data to other similar imaging modalities.

Hyperspectral imaging has been used in many fields. It can be used non-destructively to conserve, preserve and research objects of our cultural heritage, such as art and historical artifacts [6–8]. Many applications apply depth information to hyperspectral images in long range imaging, such as in aerial imaging in forestry [9] and agricultural applications. At close proximity, depth data of complex surfaces can be inferred through controlled illumination of the target and photometric stereo imaging. Skin cancer diagnosis is one medical imaging application of this setup employing a hyperspectral camera [10].

Depth imaging cameras have been used in the past to assist in segmenting objects from the background. Adding depth to hyperspectral images could benefit, e.g., in industrial robot applications where the robot has to gather information, detect and plan actions autonomously based on the sensory input. Example applications could be found for perishable products, such as in automatic fruit inventory and harvesting robots [11]. Hyperspectral imaging has previously been applied for detecting injuries in fruits [12] and with other horticultural products [13].

Combining 3D data from a Kinect V2 with hyperspectral images has previously been done in [14]. The aim of the study was to improve the accuracy of reflectance measurements for curved leaf surfaces by selecting a white reference measurement with the same height and surface normal direction as the sample. This was done by building a white reference library from measurements of a specially designed white reference sample, imaged with the same setup as the leaves.

In [15], the authors developed a 3D multiview RGB-D image-reconstruction method for imaging chlorophyl contents of tomato plants using a multispectral imager and Kinect V2. The used hyperspectral camera employed an internal scanning mechanism where the sensor is moved behind the optics. A plant was rotated around its axis while a Kinect V2 and a hyperspectral camera captured depth images and multispectral images with four selected wavelength bands. The data were used in analyzing spectral reflectance variability from different view angles and to create chlorophyl contents prediction model. The findings suggest that multiview point cloud model could produce superior plant chlorophyl measurements compared to a single-view point cloud model. The camera sensor fusion was carried out by an image registration technique based on Fourier transform, phase correlation and a rotating electric turntable with visible sticker markers.

This research demonstrates a method for fusing frame-based hyperspectral camera data with 3D depth data and an experimental application on how the depth augmented hyperspectral data can be used for measuring angle-wise reflectance of a color checker board. Comparing to the previous linescanner method described in [14], a frame-based imager imposes many benefits in terms of the ease of imaging and portability; setting up the system and capturing a scene does not require a moving linescanner. In our experiment, we selected fitting optics and the calibration method considers common reference points in calibration images and not the spectral domain, such as in the method proposed by [15]. Our imager captured hyperspectral data cubes with 133 wavelength bands. We combined them with the estimated 3D surface normals of the target object and calculated the emission angles.

Novelty of the study come from the camera fusion method of these types of cameras. The findings, challenges and topics on how this kind of data could be utilized in future research will be discussed. This kind of setup could potentially be used in, for example, imaging and researching complex surfaces for material characterization, as well as in specular reflection removal from spectra. In summary, this method provides technical support for designing and implementing a system for hyperspectral 3D point cloud creation and analysis.

2. Materials and Methods

2.1. Experimental Setup

The experimental setup consisted of a Fabry-Pérot interferometer (FPI) hyperspectral camera, Microsoft Kinect V2 depth sensing camera, two halogen lights equipped with diffusers, x-rite ColorChecker calibration board and a desk in a darkened room in Spectral imaging laboratory at University of Jyväskylä. The Kinect was aligned on top of the hyperspectral camera and attached and aligned using an assembly of a base, translation rail and mounting brackets by Thorlabs, as seen in Figure 1.

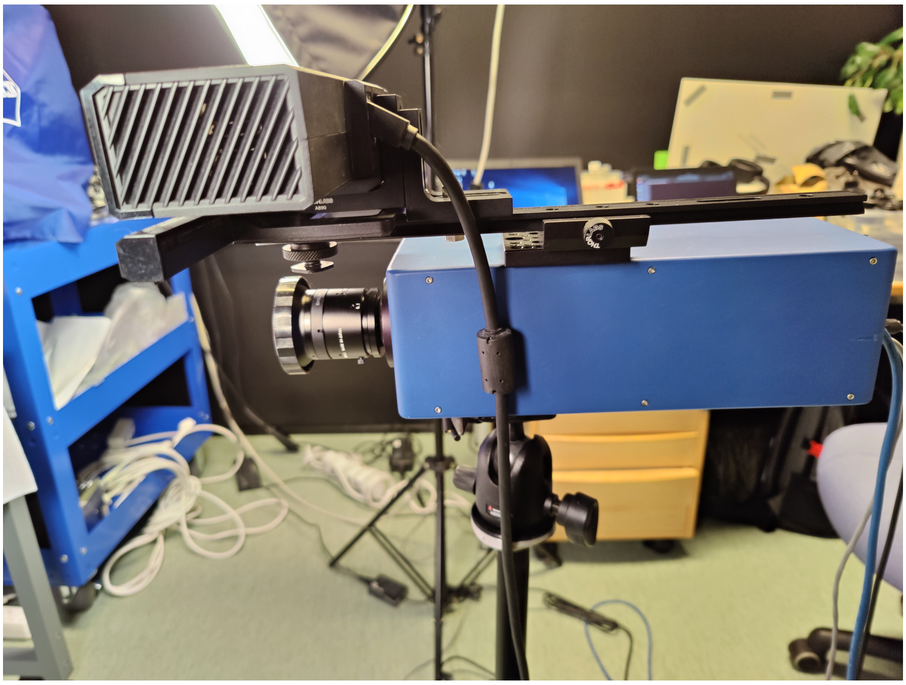


Figure 1. The prototype FPI hyperspectral camera (**below**) and the Kinect (**on top**) used in the research.

The experimental software for this study was written in Python 3.8 with OpenCV computer vision, Open3D point cloud processing, and Numpy numerical libraries. The software was targeted to work on Ubuntu Linux 20.04 LTS.

2.2. FPI Hyperspectral Camera

We used Fabry–Pérot interferometer (FPI) hyperspectral camera developed by VTT Research Centre of Finland. The camera is an assembly of optics, an interferometer, filters, and a machine vision sensor (Grasshopper3 USB3 GS3-U3-23S6C-C) with an RGB sensor. It captures a hyperspectral data cube that has (x, y) spatial dimensions and a spectral domain. The camera works by capturing multiple images and varying the interferometer settings between exposures. The Piezo-actuated interferometer consists of two metallic half-mirrors whose separation can be controlled. A beam of light entering the system interferes with itself as it reflects off the mirrors. Only integer multiples of certain wavelengths get transmitted through the mirrors [16,17].

The hyperspectral camera uses high and low-pass filters to block the unwanted wavelengths of light. Our setup used 450 nm high-pass and 850 low-pass filters, and it was calibrated to capture 80 raw bands from the calibrated 450–850 nm range. The spectral resolution (full width half maximum, FWHM) varied from 8 to 25 nm. We used CubeView [18] software to capture hyperspectral data cubes. The software converted the raw bands to 133 radiance bands using fpipy [19] Python library. The hyperspectral data were stored in fpipy defined netcdf file format with 1920 × 1200 resolution. The file size of one data cube was approximately 4.9 GB.

We aimed to capture sharp and evenly exposed images. Therefore, the aperture was set small (f/8) to have a large depth of field and to minimize vignetting that would otherwise show as a reduction in brightness towards the periphery of the image [20]. The exposure time was set to 3 s per frame to counter small aperture size and underexposed images. The total exposure time for the 80 frames of one hyperspectral image was then approximately 4 min.

2.3. Kinect V2

We used Microsoft Kinect V2 depth sensing camera for capturing depth maps of the target. Kinect V2 works by illuminating the scene with infrared light and estimates the distance to obstacles by time-of-flight (TOF) principle. The distance to obstacles is estimated measuring the time it takes light to travel from the emitter back to the infrared camera [21].

We used Libfreenect2 [22] open source driver and a modified Python wrapper based on [23]. The depth maps were captured with 512 × 424 resolution.

2.4. Cameras and Optics

Hyperspectral camera optics were selected to provide a similar field of view (FOV) to the Kinect V2, using a 100 cm working distance (WD). Since Kinect's horizontal and vertical lens opening angles were 70 and 60 degrees, the horizontal and vertical FOV was calculated as seen in Figure 2.

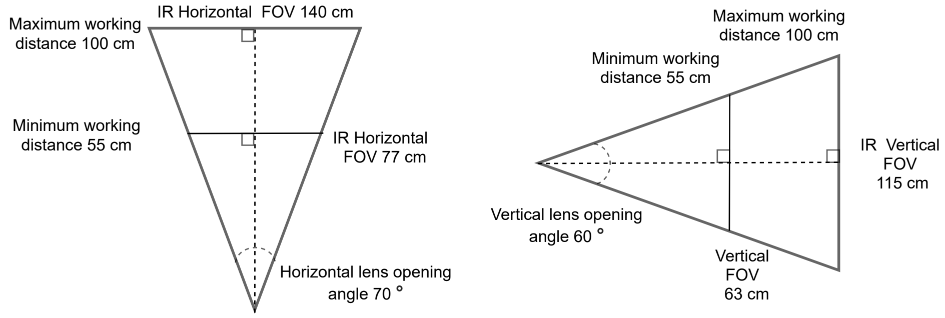

Maximum working distance 100 cm
IR Horizontal FOV 140 cm
Minimum working distance 55 cm
IR Horizontal FOV 77 cm
Horizontal lens opening angle 70 °
Maximum working distance 100 cm
Minimum working distance 55 cm
IR Vertical FOV 115 cm
Vertical lens opening angle 60 °
Vertical FOV 63 cm

Figure 2. The horizontal and vertical field of view were calculated based on the Kinect V2's lens opening angles, using 100 cm as a reference maximum working distance for the imaging setup.

The defined the required depth of field (DOF) of the target to be 45 cm (Figure 3), which determined the maximum and minimum WDs to be 100 cm and 55 cm, respectively. The resulting horizontal FOV at the minimum WD was 77 cm and 141 cm at the maximum WD.

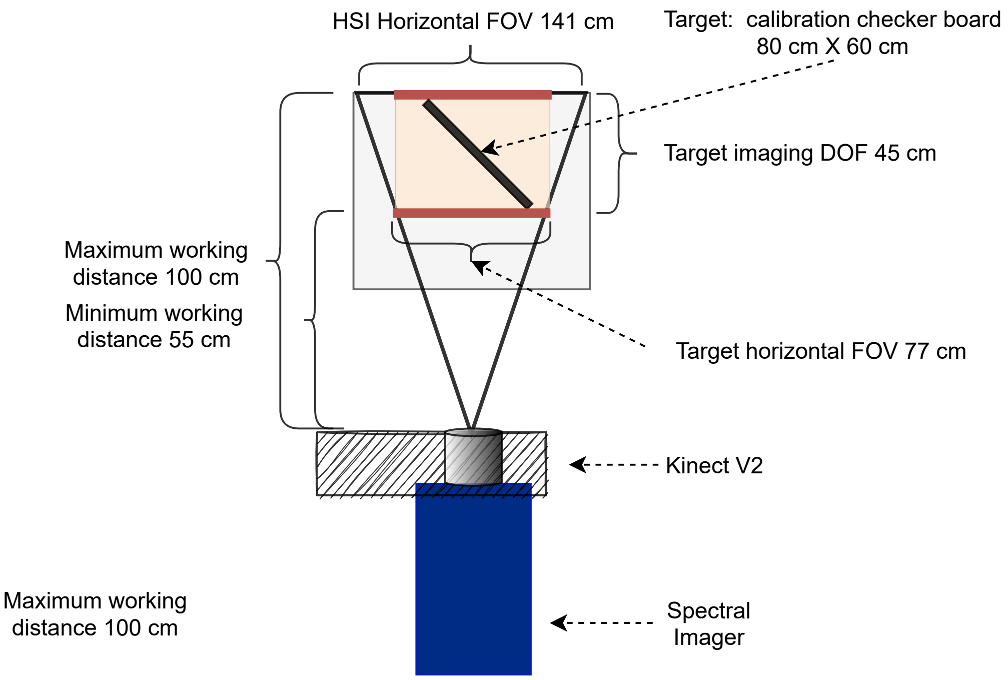

HSI Horizontal FOV 141 cm
Target: calibration checker board 80 cm X 60 cm
Target imaging DOF 45 cm
Maximum working distance 100 cm
Minimum working distance 55 cm
Target horizontal FOV 77 cm
Kinect V2
Maximum working distance 100 cm
Spectral Imager

Figure 3. Selecting optics for the target scene. A visualization of the imaging setup parameters for achieving a similar vertical and horizontal field of view with HSI and Kinect V2 sensors.

The selected lens was Basler Standard Lens (C10-0814-2M-S f8mm) with C-mount. The fixed focal length was 8.0 mm, and the resolution was 2 megapixels. With hyperspectral camera sensor, the lens provided 141 × 105 cm FOV, which is visualized with Kinect's FOV (140×115) cm in Figure 4.

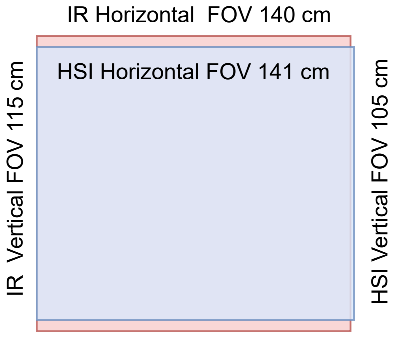


Figure 4. The hyperspectral camera lens provided a relatively similar vertical and horizontal field of views (visualized in blue) than the Kinect V2, which is visualized using red color.

By placing the Kinect on top of the hyperspectral camera and adjusting the lenses' outer surfaces to the vertically same level, we could capture hyperspectral and depth data with relatively similar parameters (Figures 3 and 4). The HSI sensor was smaller than an ideal sensor for the selected lens, but the possible vignetting effect was controlled by adjusting the iris during the acquisition.

2.5. Spectral Point Cloud Generation

In order to combine the depth data and the spectral data from Kinect and the hyperspectral camera, we need to know intrinsic camera parameters and the extrinsic camera parameters. They define the optical properties of the cameras, their relative positions and orientations to each other in the world. The data fusion of the two cameras was carried out as follows: We estimated the global point coordinates seen by Kinect, transform them to the viewpoint of the hyperspectral camera and project them onto its camera plane. Then we match the projected points to the pixels on the hyperspectral camera plane to form the spectral point cloud.

The definition of the camera matrix (also known as the camera intrinsic matrix) for both Kinect and FPI hyperspectral camera is in Equation (1), where f_x and f_y are the focal lengths in x and y directions. Correspondingly, c_x and c_y denote the principal point, which means the optical center on the sensor perpendicular to the camera's pinpoint. The parameter S is the skew [24].

$$\mathbf{K} = \begin{bmatrix} f_x & S & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}_{3\times3} \tag{1}$$

We can calculate the world to image plane transformation with full-rank (4 × 4) matrices (Equation (2)). Using full-rank matrices allows us to invert them and to calculate the image plane to the world transformation [24].

$$\begin{bmatrix} u \\ v \\ 1 \\ 1/z \end{bmatrix} = \frac{1}{z} \begin{bmatrix} \mathbf{K} & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix}_{4\times4} \begin{bmatrix} \mathbf{R} & \mathbf{T} \\ \mathbf{0} & 1 \end{bmatrix}_{4\times4} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} = \frac{1}{z} \begin{bmatrix} \mathbf{K} & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix}_{4\times4} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} \tag{2}$$

The full-rank transformation matrix $[\mathbf{R}|\mathbf{T}]_{4\times4}$ can be omitted, because the skew $S = 0$ and the camera matrix is aligned with the world. The parameters x_w, y_w and z_w are the world coordinates for a point in the point cloud. On the camera sensor plane, u, v, and z denote the camera coordinates.

The inverse camera matrix can be analytically calculated and it is defined in Equation (3).

$$\mathbf{K}^{-1} = \begin{bmatrix} 1/f_x & -S/(f_x f_y) & (Sc_y - c_x f_y)/(f_x f_y) \\ 0 & 1/f_y & -c_y/f_y \\ 0 & 0 & 1 \end{bmatrix} \tag{3}$$

Then the Kinect image plane to world transformation is defined as:

$$\begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} = z \begin{bmatrix} \mathbf{K}^{-1}_{kinect} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} \end{bmatrix}_{4\times4} \begin{bmatrix} u \\ v \\ 1 \\ 1/z \end{bmatrix} = z \begin{bmatrix} 1/f_x & 0 & -c_x f_y/(f_x f_y) & 0 \\ 0 & 1/f_y & -c_y/f_y & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \\ 1/z \end{bmatrix} \tag{4}$$

The projected coordinates from the world coordinates to the hyperspectral camera plane can be calculated with the intrinsic camera parameters $\mathbf{K}_{hyper}$ and the extrinsic parameters $[\mathbf{R}|\mathbf{T}]_{kinect\rightarrow hyper}$. The extrinsic matrix defines the transformation between the two camera locations and orientations with the rotation $\mathbf{R}$ and the translation $\mathbf{T}$ matrices.

The projection matrix of Kinect's world coordinates to the hyperspectral camera plane is defined in Equation (5).

$$\mathbf{P}_{kinect\rightarrow hyper} = \mathbf{K}_{hyper} \begin{bmatrix} \mathbf{R}_{3\times3} & \mathbf{T}_{3\times1} \end{bmatrix}_{kinect\rightarrow hyper} \tag{5}$$

The world coordinates are then projected on the hyperspectral camera sensor plane:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix}_{hyper} = \frac{1}{z_w} \mathbf{P}_{kinect\rightarrow hyper} \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix}_{kinect} \tag{6}$$

Pixels outside of the Kinect's operational range were filtered out. Hyperspectral image pixel matching is conducted by simply rounding the projected image plane coordinates to nearest even integer pixel coordinates that fit inside the Kinect image plane. The spectral point cloud is defined by (x_w, y_w, z_w) coordinates and their matching 133 spectral bands.

The current point cloud file formats do not support storing more than three color bands. Therefore, we defined a custom format using xarray [25] and netcdf [26] that contains the spatial and spectral information, point normals, and other metadata, such as the band-wise wavelengths. Xarray is a Python library that makes working with multi-dimensional data arrays with different coordinate systems easier. Netcdf is a community standard for sharing array-oriented scientific data.

2.6. FPI Hyperspectral Camera to Kinect Calibration

The intrinsic and extrinsic camera matrices can be inferred using a common and known reference image pattern. In our case, we used a 9 × 6 checkerboard image with 45 mm square size printed on standard copying paper. We captured 33 calibration images with both cameras while turning the image pattern in different angles along all axis and keeping the camera position fixed. Figure 5 depicts the calibration setup.

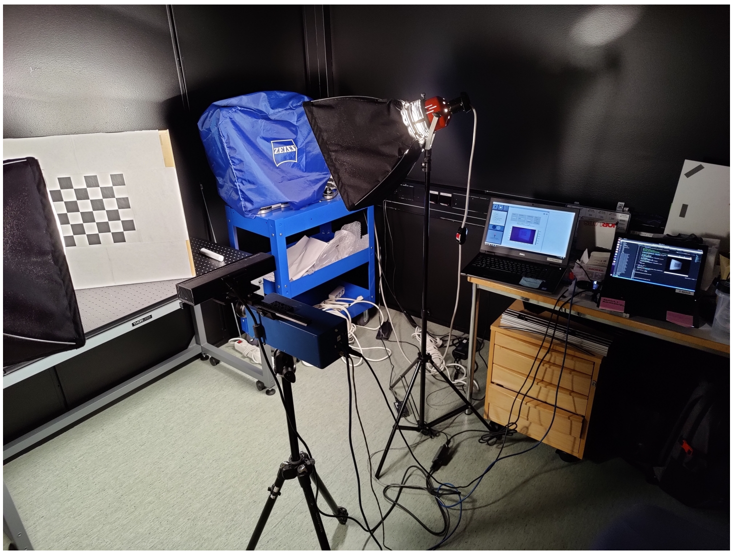


Figure 5. The experimental calibration setup with the hyperspectral camera, Kinect, halogen diffusers, and realignable checkerboard calibration pattern.

The FPI hyperspectral camera was configured to capture four images with different interferometer settings per each calibration image position. That resulted in spectral images with 8 wavelength channels. We produced the final calibration images by clipping the band values within $[0, \mu + 10\sigma]$ range to remove any outliers, such as dead pixels, averaging the bands, and normalizing them to gray scale to minimize spatial image noise. We normalized Kinect's IR images to $[0, 255]$ range and used them as-is.

We used OpenCV's `findChessboardCorners` function to automatically detect the corners of the checkerboard in the images, `cameraCalibrate` function to estimate the camera matrix (Equation (1)) parameters for both cameras and `stereoCalibrate` functions for estimating the extrinsic matrix between the two camera locations and positions with their previously determined optical properties. The resulting intrinsic parameters are listed in Table 1.

Table 1. The estimated intrinsic parameters of the cameras during the calibration.

Intrinsic Parameter	Kinect V2	FPI Hyperspectral Camera
f_x	366.261	1382.955
f_y	366.465	1383.227
c_x	255.923	1002.023
c_y	206.977	601.358

2.7. Point Cloud Registration

We used Open3D point cloud processing libraries to infer the spatial transformations between point clouds. Our experiment included five point clouds that were captured by moving the hyperspectral camera and Kinect around the target object. The point cloud of the center-most camera position was used as the target for aligning the other point clouds, which will be referred as the source point clouds.

The point clouds had to be preprocessed to filter out excessive noise. Outliers in the point clouds were identified and removed statistically based on the average distance in a neighborhood of points, as shown in Figure 6.

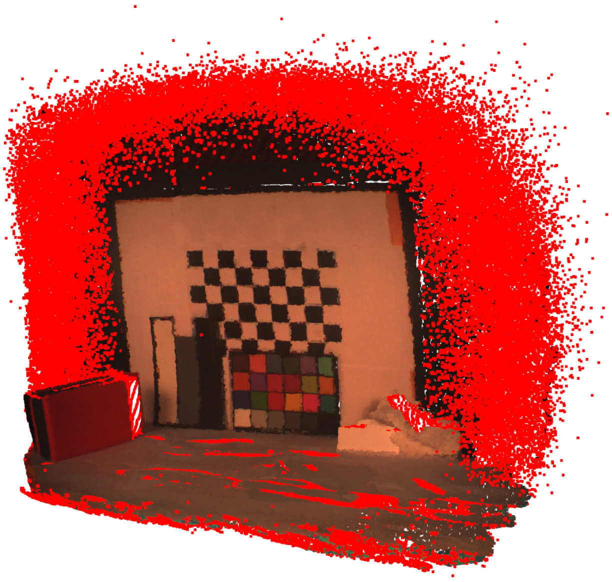


Figure 6. Visualization of the outlier removal for the point cloud in the middle camera viewpoint. The points highlighted with red were removed.

Aligning two point clouds without prior information on their initial pose in space was achieved by using global and local registration algorithms. We computed pose-invariant FPFH features (Fast Point Feature Histograms) [27] which represent the surface model properties around each point. Using FPFH speeds up the global point cloud registration significantly compared to genetic and evolutionary algorithms [27]. We downsampled the point clouds with 15 mm voxel size and estimated the point normals for each point cloud, as FPFH relies on the 3D coordinated and estimated surface normals. Open3D estimates vertex normals by calculating principal axis of the adjacent points over the closest neighboring points.

Figure 7 illustrates the registration steps. We used RANSAC [28] for the global registration. RANSAC works by picking random points from the source point cloud and finding their corresponding points in the target point cloud by querying the nearest neighbors in the FPFH feature space. A pruning step rejects false matches early. We experimentally set RANSAC pruning algorithm's correspondence distance threshold (the distance between two aligned point) to 75 mm. The algorithm's correspondence edge length was set to 0.9. It is a threshold for checking that any two arbitrary corresponding edges (line between two vertices) in the source and target point clouds are similar. The RANSAC convergence criteria was set to 400,000 iterations and 0.999 confidence.

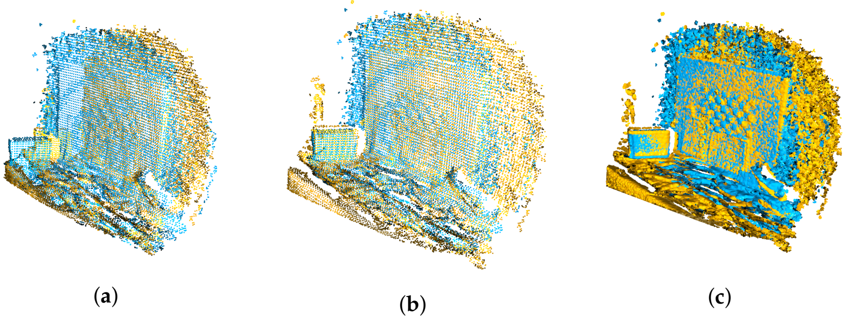


Figure 7. Visualizations of the registration of two point clouds, one shown in blue and one in yellow: (**a**) The point clouds before realignment, (**b**) after global registration, and (**c**) the refined local registration.

The next step is the local refinement with the point-to-plane ICP (Iterative Closest Point) [29] registration algorithm. We used the original outlier-filtered point clouds without downsampling and the rough transformation results from RANSAC to further refine the alignment. Figure 8 shows the fully registered point cloud with pseudo coloring and the camera viewpoints.

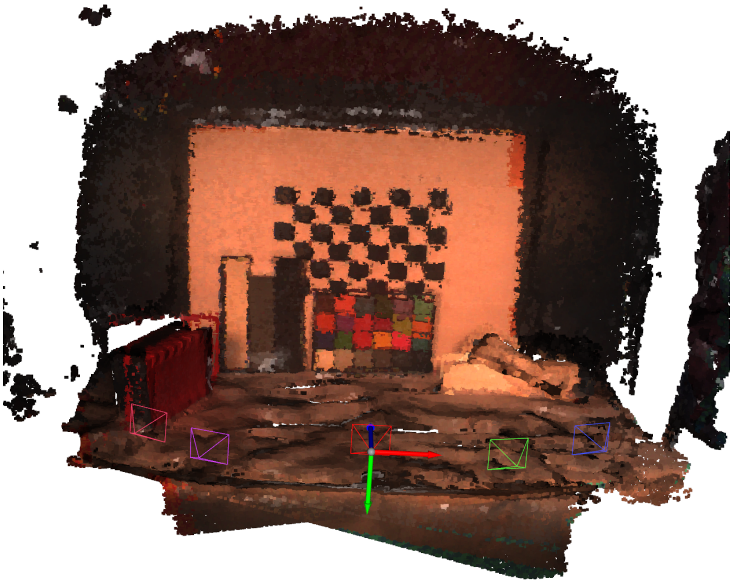


Figure 8. The fully registered point cloud with pseudo RGB coloring and the visualizations of camera capture viewpoints.

ICP produces the extrinsic transformation matrices to integrate each point cloud to the viewpoint of the central camera position. The transformation matrices also gives us the positions and orientations of each camera in relation to the central camera. We can use them later to calculate the emission angles.

2.8. Calculating Point Normals

The next step is to calculate corresponding point normals for each point in the point cloud. Normals are needed to calculate the emission angles relative to the camera positions. We used Open3D's Poisson surface reconstruction method [30] to fit a surface on the point cloud. Open3D offers functions to calculate point normals using its adjacent points. The reconstruction algorithm allows defining the depth of the underlying octree data structure. It controls the resolution of the resulting triangle mesh. We set the depth to 7, because the noise could create steep angles on flat surfaces. We applied Taubin smoothing [31] to further smoothen the fitted surface. The produced mesh is presented in Figure 9a.

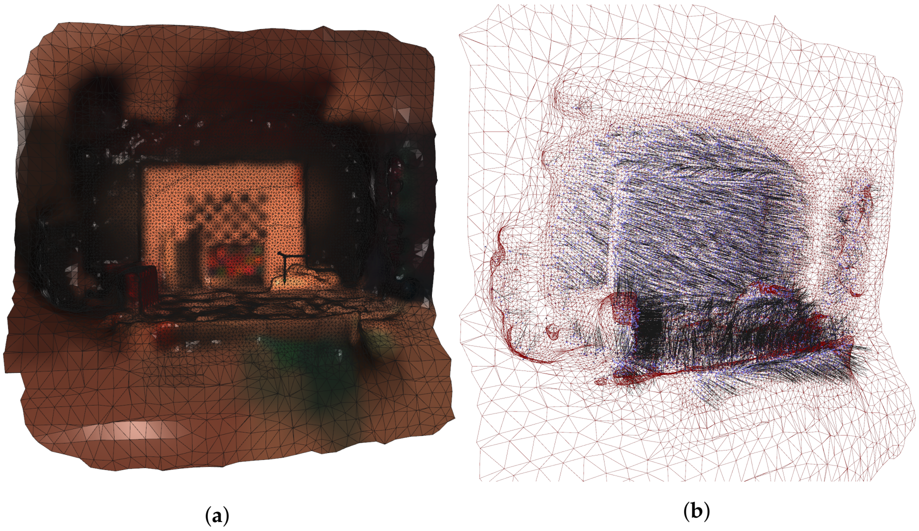

(**a**) (**b**)

Figure 9. (**a**) The fitted mesh on the fully registered point cloud with Taubin smoothing. (**b**) Visualization of the mesh triangles (red) and the points point cloud (purple) with the recalculated surface normals (black).

We assigned the normals of the closest mesh triangles to the points of the point cloud using Open3D's ray casting functions. Figure 9b illustrates a downsampled view of the new point normals showing how flat surfaces have relatively uniform normal directions.

2.9. Calculating Emission Angles

The emission angle α is the angle at which the reflected and transmitted light are received at the detector. Defining the emission angle at the surface point p then comes down to calculating the cosine between the surface normal and the vector at the direction of camera from the point p, as illustrated in Figure 10a.

The relative camera position $\vec{o}$ is acquired from the world camera translation we estimated during the registration. The translation vector T needs to be negated, because the original world-to-world transformations are defined towards the origin, the middle camera:

$$\vec{o} = (-T_x, -T_y, -T_z) \tag{7}$$

The emission angle α at the point $\vec{p}$ can be calculated as the dot product of the normal vector $\vec{n}$ and the vector $\vec{q}$ pointing from the point $\vec{p}$ towards the capturing camera position $\vec{o}$:

$$\vec{q} = \vec{o} - \vec{p} \tag{8}$$

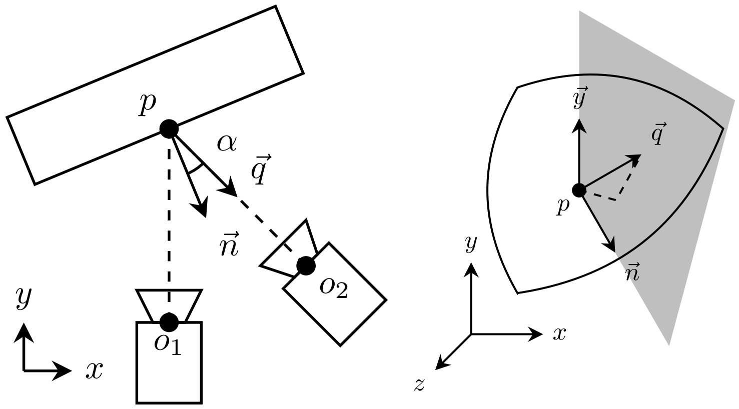
$$\alpha = \cos^{-1}\left(\frac{\vec{n} \cdot \vec{q}}{\|\vec{n}\| \, \|\vec{q}\|}\right) \tag{9}$$

The resulting emission angle is defined in $[0, 90]$ degree range. Our camera is only moved in xz-axis and the position in y axis is kept relatively fixed with approximately

2 cm variation between capture positions. Therefore, we calculate the signs of the emission angles to have $[-90, 90]$ degree range. We split the space with vertical plane along the direction of the point normal $\vec{n}$ and the y axis.

$$\alpha_{signed} = \alpha \operatorname{sign}((\vec{n} \times \vec{y}) \cdot \vec{q}) \tag{10}$$

The sign of the dot product between the splitting plane and camera pointing vector $\vec{q}$ determines on which side of the plane point $\vec{p}$ is located. Figure 10b depicts the splitting plane and the related vectors. Equation (10) gives us the signed emission angle.



(**a**) (**b**)

Figure 10. (**a**) Emission angle calculation for the point p. (**b**) Illustration of the vertical plane that splits the space along the direction of the point normal and the y axis. It is used for determining the sign of the emission angle.

2.10. Reflectance and Its Angular Dependence

Reflectance is defined as a material's ability to reflect incoming electromagnetic radiation. Reflectance is a unitless quantity between zero and one; a material with reflectance of one will reflect all radiation incident on it, and a material with reflectance of zero will not reflect anything. In this work, the quantity of interest is spectral reflectance, a set of reflectances each corresponding to a wavelength channel. In addition to wavelength, reflectance can depend on the directions of incident and reflected light [32].

Reflection can be divided into specular reflection from an optically smooth surface, such as a mirror, and diffuse reflection from a rough surface such as soil. Reflections from real surfaces are often a mix of these two. For example, a body of water will reflect the image of a light source in one direction, and in another direction appear the color of the solids suspended in the water [32].

The simplest analytical expression for reflection from diffuse surfaces is known as Lambert's law. The law is based on the observation that the apparent brightness of a surface is independent of the angle it is viewed from. Lambert's law states that the only directional dependence to the intensity of reflected light comes from the incidence angle, as this affects the intensity of incident light. Although the reflections of real surfaces are not perfectly Lambertian, some bright surfaces come close [32].

To find the spectral reflectance of a surface, one must quantify both the light reflected from the surface and the light arriving to it. This is often done by measuring the unknown surface along with a standard that has known reflectance properties. If the standard is assumed to be perfectly white, i.e., it reflects all light arriving to it in the wavelength region of the measurement, the spectral reflectance R is given by:

$$R = \frac{I}{I_{white}}, \tag{11}$$

where I is the spectral radiance reflected from the target, and I_{white} is the spectral radiance reflected from the white reference target [32]. A similar approach was taken to calculate spectral reflectance from our measurements. The used hyperspectral camera recorded a spectral radiance I for each of its pixels. A white reference measurement was made by placing a block of Spectralon [33] reference material in the imaged scene. Spectralon is a common reflectance standard that is highly reflecting and diffuse in our spectral range extending from visual wavelengths to the shorter end of near-infrared. The reference radiance I_{white} was calculated by averaging the spectral radiance over the reference target area.

The lamps used to illuminate the scene were positioned and aligned on both sides of the camera so that the specular reflections were minimized to the front view of the color checker board. With this lighting geometry, the measured reflectances should show higher values when the target was imaged at a side view. The target is expected to have a specular reflection component in its reflection.

3. Results

Figure 11 depicts the measurement results for each tile in the color checker. Each tile has been measured with the experimental setup from five different angles and their average spectra were plotted in the top and the intensity histograms, the band-wise sum of the reflectances, per emission angle on bottom, respectively. Each tile was cropped by hand from the fully registered point cloud.

The tiles on the color checker board in Figure 12 correspond to the plots in Figure 11. The measured emission angles on the checker board varied between -28 and 22 degrees. The results verify that the color checker tiles are mostly diffuse surfaces and the emission angle has little effect to the intensity. Some liminal intensity attenuation can be observed around zero emission angle. This is expected as the positioning of the lamps cause the front view have dimmer illumination. The side views receive more light due to specular reflections.

In Figure 11, the red plots illustrate the average measured spectrum from the front view. It represents the spectrum of the original hyperspectral image without emission angle information. The blue plot is the average over all measurements in different angles. The gray color in the plots illustrates the band-wise standard deviation. Some color tiles showed larger band-wise intensity fluctuations. More fluctuations are observed at the last bands. This may due to the fact that this type of camera has previously been observed to produce noise in large wavelengths. However, we can see more deviance depending on the tile color. For example, the most reflective tile s (white) shows more noise than the darker toned tiles t–x. In tiles c, f, and r we noticed fluctuations in the smaller wavelengths, and for j, l, o, and p in the middle wavelengths. Common for all the noisiest wavebands is that they all share large intensity values. The red plots shows that for most of the colors, the spectrum does not change much between different views, as could be seen from the intensity histograms as well. The results suggest that the spectra from different angles are similar as in the original front view hyperspectral image.

In Table 2, we list the RMSE (root mean squared error) values and their standard deviations for each tile. The errors are calculated between the mean spectra of the central camera view and the spectra from other view positions. The point of this measurement is to quantify how much the emission angle affects the measured spectra. The deviation values ranged from approximately 0.01 to 0.05, which can be regarded as small, as could be stated based on the intensity histograms. We used cosine, sometimes called the spectral angle [34] in the spectral domain, to measure the differences in spectrum shape. The differences are between 1.5 and 3.2 degrees, which would refer to emission angle having little effect on the shape of the spectrum with these targets.

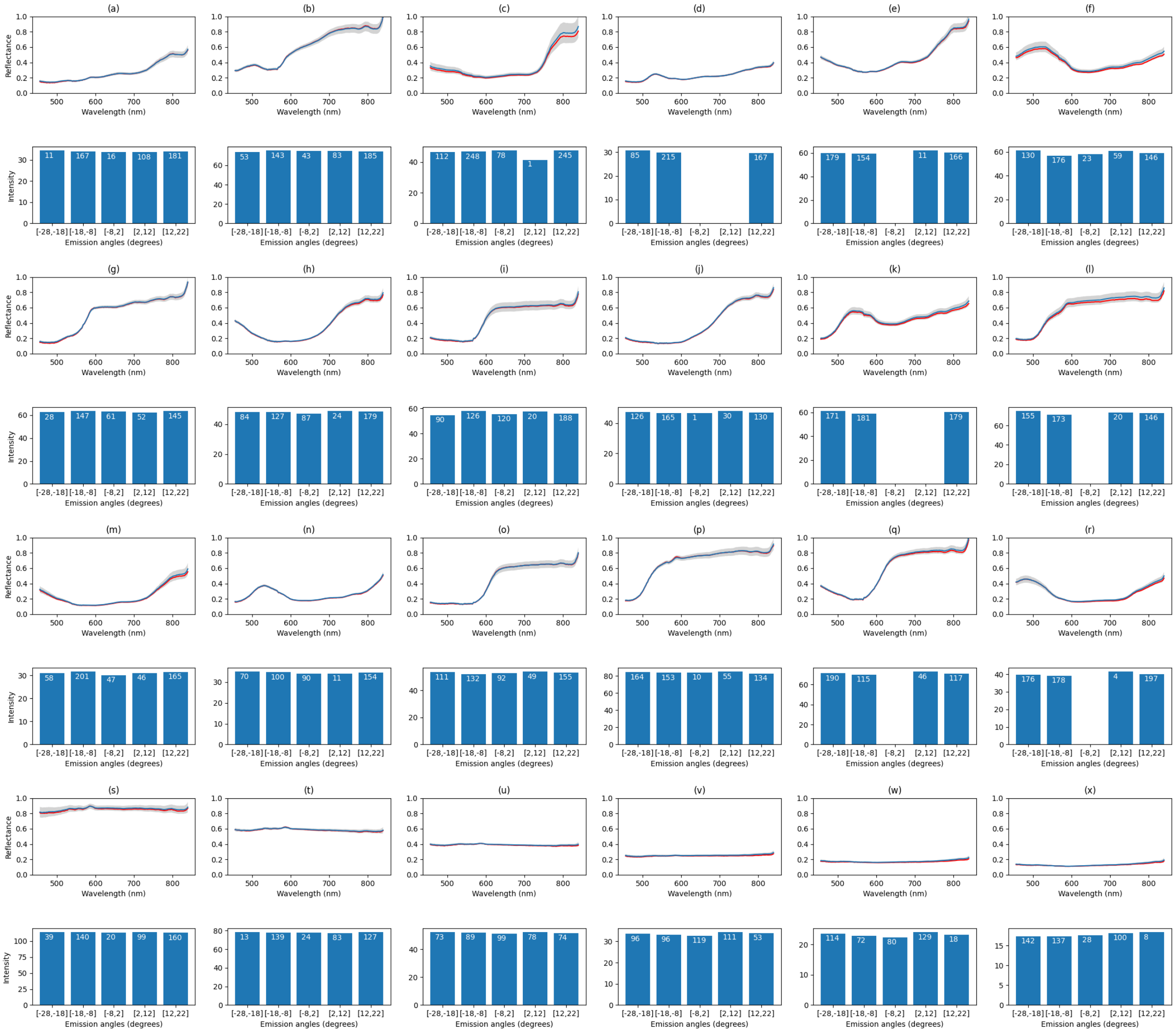


Figure 11. The average spectra and the intensity histograms of each matching color tile (**a**–**x**) of the reference color checker in Figure 12. The average spectra of all measurements are plotted on top in blue, the front view average spectrum is plotted in red and the intensity histograms, the band-wise sum of the reflectances, per emission angle are on the bottom. The sample count per a histogram bin is displayed in white. The gray color in the top plots illustrate the band-wise standard deviations.

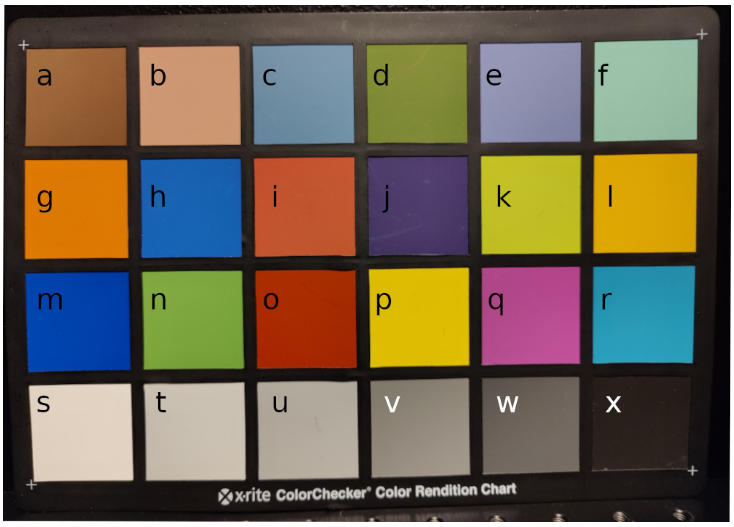


Figure 12. The color checker board used for calculating the result reflectances. Letters a–x correspond to each tile and a result plot in Figure 11.

Table 2. Root mean squared differences and spectral angles (cosine) of measured spectra for each corresponding color checker tile, in Figure 12, compared to the averaged front view spectrum from central camera position.

	a	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**i**	**j**	**k**	**l**
RMSE	0.0248	0.0427	0.0493	0.0125	0.0301	0.0339	0.0273	0.0219	0.0474	0.0222	0.0316	0.0413
std	0.0039	0.0093	0.0136	0.0004	0.0050	0.0069	0.0039	0.0029	0.0133	0.0028	0.0065	0.0105
cos	2.4171	1.5101	2.4386	2.1830	1.7203	1.7153	1.5530	1.7175	1.7402	1.5635	1.6439	1.4640
std	1.5109	0.8921	1.8757	0.4083	0.9819	1.2294	0.7569	0.8792	1.8425	0.5303	1.1196	0.8974
	m	**n**	**o**	**p**	**q**	**r**	**s**	**t**	**u**	**v**	**w**	**x**
RMSE	0.0259	0.0121	0.0397	0.0416	0.0313	0.0311	0.0338	0.0200	0.0121	0.0145	0.0108	0.0096
std	0.0035	0.0011	0.0097	0.0081	0.0067	0.0058	0.0047	0.0016	0.0003	0.0008	0.0003	0.0006
cos	2.8011	1.9333	1.6055	1.4051	1.3058	2.4644	1.2502	1.3298	1.5481	1.9872	2.5215	3.1784
std	2.4471	0.6890	1.2743	0.9680	0.6200	2.2738	0.8716	0.5919	0.2548	0.3869	0.4246	0.9830

We kept the lighting setup fixed and, as seen from the results, we observed only small fluctuations in the intensity values while moving the camera. Based on a visual inspection the colorchecker board appeared diffuse, and, as such, could be approximated as Lambertian. The intensity of light reflected from Lambertian surfaces is not dependent on the emission angle and, thus, we would expect to see no variation in intensities measured from different angles. The results of the experiment are in line with this expectation.

In Table 3, we list registration errors of the individual point clouds that were used for creating the final dense point cloud. The correspondence set size is the number of point pairs that have correspondence to each each other in the source and target point clouds. The fitness is the number of inlier correspondences divided by the number of points in the target point cloud. A larger value is better and it means that the point clouds have more overlap. The RMSE (root mean squared error) is calculated over all the inlier correspondences. Each pruned point cloud had approximately 140,000 points. The quite high correspondence set size (approx. 40,000–50,000) and fitness (0.29–0.36) values suggest that the registered point clouds had a lot of overlap which made the resulting registration quite reliable as we can visually confirm on the resulting full point cloud in Figure 8. However, we can see some off-alignment on the checkerboard.

Table 3. The refined local registration errors of the listed source point clouds (side views) to the target camera position (center).

Viewpoint	Fitness	Inlier RMSE	Correspondence Set Size
Left	0.356	1.975	50,192
Left-most	0.296	1.989	41,620
Right	0.358	2.005	49,098
Right-most	0.2901	2.002	40,006

4. Discussion

Figure 8 illustrates the final point cloud with pseudo coloring. Due to inaccuracies in the Kinect-to-FPI hyperspectral camera calibration, some transformations are off by 1–2 cm at most. The most significant error source was the sub-optimal mounting of Kinect. Small deviations in the orientation between the two cameras caused large errors in extrinsic parameter calculations, as the camera had to be moved to capture images on multiple angles.

Originally, the hyperspectral camera used optics that had a small field of view, for which OpenCV could not satisfactorily solve the extrinsic parameters. Therefore, we selected lens that matches closely the field of view of Kinect. The calibration would have benefited from averaging multiple infra-red pictures from the Kinect to reduce noice.

We observed that the point clouds captured by Kinect gauged depth values depending on the brightness of the target, darker areas gaining shorter distances than the brighter areas. This is especially visible in the checkerboard pattern in Figure 7c. The intensity related error is known to occur with Kinect V2 [35]. We fitted a mesh on the plane and recalculated the point normals to diminish these alterations to the emission angles of the spectra. One future improvement would be matching the point cloud normals by ray tracing and finding the intersecting mesh triangles.

Looking at Figure 11, we see a spike in the average spectra around infrared range. This is expected as the Kinect V2 illuminates the scene and it is detected by the hyperspectral camera. This should be taken into account when using this kind of time-of-flight depth camera, if the application operates around these wavelengths.

To incorporate more accurate and dense spectral point cloud, using the rest of the spectra in hyperspectral data cubes should be considered. In the presented implementation, only the closest spatially matching spectra in the Kinect's perspective were considered and the rest were pruned.

A previous study by [14] showed how linescanner hyperspectral camera and Kinect V2 can be used to create a 3D white referencing library which offers tilt angle specific white references for hyperspectral calibration. The motivation of the study was to improve the calibration of soy bean leaf images. The authors used a ball shaped white reference in creating the 3D white referencing library, which is something we should consider using in our emission angle dependent reflectance measurements. The authors detected a significant difference at the angled reflectance calibration compared to a flat reference. We used an averaged white reference spectrum from a flat reference object over multiple angles. In the study, the authors envisioned using LiDAR sensors in 3D scanning and use it in field environments. The strength of LiDAR is in long distance applications. The presented method should work in mid-range complex surface imaging applications and be more portable in comparison.

Future research topics would include fitting a model, such as a neural network, with the data produced by the system and use it to interpolate the emission angle dependent reflectance of glossier materials. Using a lighting setup, such as the one presented in this research, we could potentially infer the reflectance at an emission angle where specular reflections are minimized. Specular reflections occur on glossy surfaces.

Exchanging the two light sources of the experimental setup for just one with a collimated output and tracking the relative position of the camera system and the light source would allow inferring the incidence angle of light from the 3D point cloud. With both incidence and emission angles and spectral data recorded for each pixel of the hyperspectral image, the method could be used to measure a spectral version of the bidirectional reflectance distribution function (BRDF). BRDF is a reflectance quantity that is related to the changes in reflectance with different incidence and emission angles of light. BRDF can be used for material characterization and, for example, producing digital textures. Typically measurements of this quantity for a sample require maneuvering either a light source and a detector [36] or the sample itself [37] to accurately to measure it with a series of incidence and emission angles. With the setup described in this study, including the light source upgrade, one could determine the BRDF of a material with very few measurements taken of a rounded sample. The curved surface would include a wide array of incidence and emission angles, possibly enough to construct a spectral BRDF for a material from just one capture.

5. Conclusions

We demonstrated a sensor fusion method for combining data from frame-based hyperspectral and a depth camera. We created an experimental application on how to utilize the depth augmented hyperspectral data to measure emission angle dependent reflectance from a multi-view inferred point cloud.

The method could successfully combine the 3D point cloud data and hyperspectral data from different viewpoints. The calculated angle dependent reflectance results refer that the target color checker board has Lambertian surface properties. The significance of this study is in the remarks and implementation details of designing a system for an imaging application augmenting frame-based hyperspectral data with time-of-flight depth camera data, as well as in the future research ideas we presented in the discussion chapter.

Author Contributions: Conceptualization, S.R.; methodology, S.R., L.L., S.K. and A.-M.R.-H.; software, S.R.; writing—original draft preparation, S.R. and L.L.; writing—review and editing, S.R., A.-M.R.-H., I.P., L.L. and S.K.; visualization, S.K.; supervision, S.R. and I.P.; project administration, I.P.; funding acquisition, I.P. All authors have read and agreed to the published version of the manuscript.

Funding: The work is related to the iADDVA—Adding Value by Creative Industry Platform project that has received funding from Council of Tampere Region (Decision number: A77069) and European Regional Development Fund React-EU (2014–2023) and Leverage from the EU 2014–2020. This project has been funded with support from the European Commission. This publication reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are openly available in Zenodo at [https://doi.org/10.5281/zenodo.7108216, accessed on 1 October 2022].

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

BRDF	Bidirectionald reflectance distribution function
DOF	Depth of field
FOV	Field of view
FPFH	Fast point feature histogram
FPI	Fabry–Pérot interferometer
HSI	Hyperspectral Imaging
RANSAC	Random sample and consensus
WD	Working distance

References

1. Lillesand, T.; Kiefer, R.; Chipman, J. *Remote Sensing and Image Interpretation*, 6th ed.; John Wiley & Sons: Hoboken, NJ, USA, 2007. [CrossRef]
2. Choubik, Y.; Mahmoudi, A. Machine Learning for Real Time Poses Classification Using Kinect Skeleton Data. In Proceedings of the 2016 13th International Conference on Computer Graphics, Imaging and Visualization (CGiV), Beni Mellal, Morocco, 29 March–1 April 2016; pp. 307–311. [CrossRef]
3. El-laithy, R.A.; Huang, J.; Yeh, M. Study on the use of Microsoft Kinect for robotics applications. In Proceedings of the 2012 IEEE/ION Position, Location and Navigation Symposium, Myrtle Beach, SC, USA, 23–26 April 2012; pp. 1280–1288. [CrossRef]
4. Rao, D.; Le, Q.V.; Phoka, T.; Quigley, M.; Sudsang, A.; Ng, A.Y. Grasping novel objects with depth segmentation. In Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, Taipei, Taiwan, 18–22 October 2010; pp. 2578–2585. [CrossRef]
5. Saari, H.; Pölönen, I.; Salo, H.; Honkavaara, E.; Hakala, T.; Holmlund, C.; Mäkynen, J.; Mannila, R.; Antila, T.; Akujärvi, A. Miniaturized hyperspectral imager calibration and UAV flight campaigns. In *Sensors, Systems, and Next-Generation Satellites XVII*; SPIE: Bellingham, WA, USA, 2013; Volume 8889, pp. 448–459.
6. Striova, J.; Dal Fovo, A.; Fontana, R. Reflectance imaging spectroscopy in heritagescience. *La Rivista del Nuovo Cimento* **2020**, *43*, 515–566. [CrossRef]
7. Bayarri, V.; Sebastián, M.A.; Ripoll, S. Hyperspectral Imaging Techniques for the Study, Conservation and Management of Rock Art. *Appl. Sci.* **2019**, *9*, 5011. [CrossRef]
8. Sandak, J.; Sandak, A.; Legan, L.; Retko, K.; Kavčič, M.; Kosel, J.; Poohphajai, F.; Diaz, R.H.; Ponnuchamy, V.; Sajinčič, N.; et al. Nondestructive Evaluation of Heritage Object Coatings with Four Hyperspectral Imaging Systems. *Coatings* **2021**, *11*, 244. [CrossRef]
9. Pölönen, I.; Annala, L.; Rahkonen, S.; Nevalainen, O.; Honkavaara, E.; Tuominen, S.; Viljanen, N.; Hakala, T. Tree Species Identification Using 3D Spectral Data and 3D Convolutional Neural Network. In Proceedings of the 2018 9th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS), Amsterdam, The Netherlands, 23–26 September 2018; pp. 1–5. [CrossRef]
10. Lindholm, V.; Raita-Hakola, A.M.; Annala, L.; Salmivuori, M.; Jeskanen, L.; Saari, H.; Koskenmies, S.; Pitkänen, S.; Pölönen, I.; Isoherranen, K.; et al. Differentiating Malignant from Benign Pigmented or Non-Pigmented Skin Tumours; A Pilot Study on 3D Hyperspectral Imaging of Complex Skin Surfaces and Convolutional Neural Networks. *J. Clin. Med.* **2022**, *11*, 1914. [CrossRef] [PubMed]
11. Tang, Y.; Chen, M.; Wang, C.; Luo, L.; Li, J.; Lian, G.; Zou, X. Recognition and Localization Methods for Vision-Based Fruit Picking Robots: A Review. *Front. Plant Sci.* **2020**, *11*, 510. [CrossRef] [PubMed]
12. Pan, L.; Zhang, Q.; Zhang, W.; Sun, Y.; Hu, P.; Tu, K. Detection of cold injury in peaches by hyperspectral reflectance imaging and artificial neural network. *Food Chem.* **2016**, *192*, 134–141. [CrossRef] [PubMed]
13. Lu, Y.; Saeys, W.; Kim, M.; Peng, Y.; Lu, R. Hyperspectral imaging technology for quality and safety evaluation of horticultural products: A review and celebration of the past 20-year progress. *Postharvest Biol. Technol.* **2020**, *170*, 111318. [CrossRef]
14. Zhang, L.; Jin, J.; Wang, L.; Huang, P.; Ma, D. A 3D white referencing method for soybean leaves based on fusion of hyperspectral images and 3D point clouds. *Precis. Agric.* **2020**, *21*, 1173–1186. [CrossRef]
15. Sun, G.; Wang, X.; Sun, Y.; Ding, Y.; Lu, W. Measurement Method Based on Multispectral Three-Dimensional Imaging for the Chlorophyll Contents of Greenhouse Tomato Plants. *Sensors* **2019**, *19*, 3345. [CrossRef] [PubMed]
16. Eskelinen, M.A. Computational Methods for Hyperspectral Imaging Using Fabry–Perot Interfer-Ometers and Colour Cameras. Ph.D. Thesis, University of Jyväskylä, Jyväskylä, Finland, 2019.
17. Saari, H.; Aallos, V.V.; Akujärvi, A.; Antila, T.; Holmlund, C.; Kantojärvi, U.; Mäkynen, J.; Ollila, J. Novel miniaturized hyperspectral sensor for UAV and space applications. In *Sensors, Systems, and Next-Generation Satellites XIII*; Meynart, R., Neeck, S.P., Shimoda, H., Eds.; International Society for Optics and Photonics; SPIE: Bellingham, WA, USA, 2009; Volume 7474, p. 74741M. [CrossRef]
18. Trops, R.; Hakola, A.M.; Jääskeläinen, S.; Näsilä, A.; Annala, L.; Eskelinen, M.A.; Saari, H.; Pölönen, I.; Rissanen, A. Miniature MOEMS hyperspectral imager with versatile analysis tools. In *MOEMS and Miniaturized Systems XVIII*; SPIE: Bellingham, WA, USA, 2019; Volume 10931, pp. 204–211.
19. Eskelinen, M.A.; Hämäläinen. Fpipy Python Library. Available online: https://github.com/silmae/fpipy (accessed on 19 September 2022).
20. Greivenkamp, J.E. *Field Guide to Geometrical Optics*; SPIE Press: Bellingham, WA, USA, 2004; Volume 1.
21. Sell, J.; O'Connor, P. The Xbox One System on a Chip and Kinect Sensor. *IEEE Micro* **2014**, *34*, 44–53. [CrossRef]
22. Xiang, L.; Echtler, F.; Kerl, C.; Wiedemeyer, T.; Lars; Zou, H.; Gordon, R.; Facioni, F.; Wareham, R.; Goldhoorn, M.; et al. libfreenect2: Release 0.2. Open source drivers for the Kinect for Windows v2 device. *Zenodo* **2016**. [CrossRef]
23. pykinect2 Libfreenect2 Python Wrapper. GitHub Repository. Available online: https://github.com/kiddos/pykinect2 (accessed on 19 September 2022).
24. Szeliski, R. *Computer Vision: Algorithms and Applications*; Springer Nature: Berlin/Heidelberg, Germany, 2022.
25. Hoyer, S.; Hamman, J. xarray: N-D labeled arrays and datasets in Python. *J. Open Res. Softw.* **2017**, *5*, 10. [CrossRef]

26. Network Common Data Form (NetCDF). Available online: https://www.unidata.ucar.edu/software/netcdf/ (accessed on 27 September 2022).
27. Rusu, R.B.; Blodow, N.; Beetz, M. Fast Point Feature Histograms (FPFH) for 3D registration. In Proceedings of the 2009 IEEE International Conference on Robotics and Automation, Kobe, Japan, 12–17 May 2009; pp. 3212–3217. [CrossRef]
28. Fischler, M.A.; Bolles, R.C. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Commun. ACM* **1981**, *24*, 381–395. [CrossRef]
29. Chen, Y.; Medioni, G. Object modeling by registration of multiple range images. In Proceedings of the 1991 IEEE International Conference on Robotics and Automation, Sacramento, CA, USA, 9–11 April 1991; Volume 3, pp. 2724–2729. [CrossRef]
30. Kazhdan, M.; Bolitho, M.; Hoppe, H. Poisson Surface Reconstruction. In Proceedings of the Symposium on Geometry Processing, Sardinia, Italy, 26–28 June 2006; Sheffer, A., Polthier, K., Eds.; The Eurographics Association: Eindhoven, The Netherlands, 2006. [CrossRef]
31. Taubin, G. Curve and surface smoothing without shrinkage. In Proceedings of the IEEE International Conference on Computer Vision, Cambridge, MA, USA, 20–23 June 1995; pp. 852–857. [CrossRef]
32. Hapke, B. *Theory of Reflectance and Emittance Spectroscopy*, 2nd ed.; Cambridge University Press: Cambridge, UK, 2012. [CrossRef]
33. Spectralon Diffuse Reflectance Material. Available online: https://www.labsphere.com/product/spectralon-diffuse-reflectance-material/ (accessed on 21 September 2022).
34. Neware, R.; Khan, A. Identification of agriculture areas in satellite images using Supervised Classification Technique. *J. Creat. Behav.* **2018**, *6*, 682–688.
35. Sarbolandi, H.; Lefloch, D.; Kolb, A. Kinect range sensing: Structured-light versus Time-of-Flight Kinect. *Comput. Vis. Image Underst.* **2015**, *139*, 1–20. [CrossRef]
36. Li, H.; Chen, M.; Deng, C.; Liao, N.; Rao, Z. Versatile four-axis gonioreflectometer for bidirectional reflectance distribution function measurements on anisotropic material surfaces. *Opt. Eng.* **2019**, *58*, 124106. [CrossRef]
37. Dana, K.J.; Ginneken, B.V.; Nayar, S.K.; Koenderink, J.J. Reflectance and Texture of Real-World Surfaces. *ACM Trans. Graph.* **1999**, *18*, 34. [CrossRef]