UNIVERSIDAD DE GRANADA   SIBERIAN FEDERAL UNIVERSITY

СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ | SIBERIAN FEDERAL UNIVERSITY

Plant Species Detection in Aerial and Satellite Images using Deep Learning

Agreement for *Co-tutelle* of PhD Thesis
Leading to the award of a Dual doctoral Degree
between
Siberian Federal University
and
University of Granada
within the framework of
PhD 09.06.01 "Informatics and computer facilities",
05.13.17 "Theoretical Foundations of Informatics" (SFU);
PhD B25.56.1 "Information and Communication Technologies" (UGR)

by

Anastasiia Safonova

Granada, May 2021

Editor: Universidad de Granada. Tesis Doctorales
Autor: Anastasiia Safonova
ISBN: 978-84-1306-938-8
URI: http://hdl.handle.net/10481/69661

This doctoral thesis has been supported by the Erasmus+ Program of the European Union at the University of Granada, Granada, Spain (Project No. KA107) in 2017/2018 and the Grant of the President of the Russian Federation for studying and research internships abroad (Project No. 05-1997) in 2018/2019.

Table of contents

Acronyms

DL	Deep Learning
CNN	Convolutional Neural Network
DNN	Deep Neural Networks
RS	Remote Sensing
UAV	Unmanned Aerial Vehicles
RGB	Red, Green, Blue
NIR	Near Infrared
NDVI	Normalized Difference Vegetation Index
GNDVI	Green Normalized Difference Vegetation Index
ITC	Individual Tree Crowns
VGG	Visual Geometry Group
R-CNN	Regional Convolutional Neural Network

Abstract

It is unquestionable that the conservation of plant species is essential for the preservation of nature, climate, and human wellbeing. Classically, the task of conserving threatened plant species was generally done through direct field supervision by natural resources managers. Thanks to the advances in sensor technologies and also in aircrafts such as unmanned aerial vehicles (UAV), high resolution remote sensing (RS) data are becoming an important resource for monitoring threatened plant species in large areas. Such RS data are usually multi-spectral images, with three or more bands, of up to 3 cm/pixel resolution, providing an orthogonal or quasi-orthogonal view of the considered plant species. This information may not be as complete as the information provided by natural images; however, it might be sufficient to monitor tree species located in very large areas with difficult access.

The task of analyzing RS images is usually performed using classical algorithms that require a high level of human intervention. In the last ten years, Deep Learning (DL) models in general and deep Convolutional Neural Networks (CNNs) in particular have shown impressive results in extracting spatial patterns from natural images. Indeed, CNNs constitute the state-of-the-art in all computer vision tasks, in image classification, object detection, and segmentation. Nevertheless, the potential of deep CNNs have not been fully explored in high resolution orthogonal and quasi-orthogonal images, especially in plant species conservation.

This thesis presents one of the first studies in exploring the potential of deep CNNs, data preprocessing and high resolution RS data, in addressing plant species conservation problems. In particular, this thesis presents the results and analysis of deep CNN models in three different problems from natural sciences:

1. The detection of Fir trees (*Abies Sibiric*a) damaged by the bark beetle in UAV images using DL.

2. The estimation of olive tree biovolume from UAV multi-resolution image segmentation using Mask R-CNN.

3. The detection of Spruce trees (*Picea Abies*) infected by bark beetle in UAV images using YOLOs architectures.

The main objective of this thesis is to develop robust and accurate DL models for the monitoring of different plant species using UAV images. The particular objectives to achieve the main objective are:

- To build three high-quality datasets for each one of the three considered problems.

- To design the appropriate pre-processing methods that reduce noise and uncertainty in the features and annotations.

- To develop robust and accurate CNN-based models for each case study.

The results of the first two chapters of this thesis have been published in two journals ranked as Q1 and Q2 in JCR. The results of the third chapter have been submitted to “IEEE Transactions on Geoscience and Remote Sensing”.

This thesis is organized into five chapters. The first chapter introduces the considered problems, background, and objectives of the thesis. Chapter two presents the built dataset, detection model and pre-processing technique to address the detection of Fir trees (*Abies Sibirica*) damaged by the bark beetle in UAV images. Chapter three presents the built dataset, segmentation model and pre-processing techniques to estimate olive tree biovolume from UAV multi-resolution image segmentation. Chapter four gives the built dataset, models, and pre-processing technique to address the detection of Spruce trees (*Picea Abies*) infected by bark beetle in UAV images using YOLOs architectures. Finally, Chapter five provides conclusions and future work.

Resumen

Es indiscutible que la conservación de las especies de plantas es esencial para la conservación de la naturaleza, el clima y el bienestar humano. Clásicamente, la tarea de conservar las especies vegetales amenazadas se realizaba generalmente mediante una supervisión directa sobre el terreno por parte de los gestores de los recursos naturales. Sin embargo, con los avances en las tecnologías de sensores y también de los vehículos aéreos no tripulados (UAV), los datos de teledetección de alta resolución se están convirtiendo en un recurso importante para el seguimiento de las especies de plantas amenazadas en grandes áreas. Estos datos de teledetección suelen ser imágenes multiespectrales, con tres o más bandas, de hasta 3 cm/píxel de resolución, que proporcionan una visión ortogonal o casi-ortogonal de la especie vegetal considerada. Esta información puede no ser tan completa como la que proporcionan las imágenes naturales. Sin embargo, puede ser suficiente para vigilar especies arbóreas situadas en zonas muy extensas y de difícil acceso.

La tarea de analizar las imágenes de teledetección suele realizarse mediante algoritmos clásicos que requieren un alto nivel de intervención humana. En los últimos diez años, los modelos de aprendizaje profundo en general y las redes Neuronales Convolucionales (CNNs) en particular han mostrado resultados impresionantes en la extracción de patrones espaciales de imágenes naturales. De hecho, las CNNs constituyen el estado del arte en todas las tareas de visión por computador, en clasificación de imágenes, detección de objetos y segmentación. Sin embargo, el potencial de las CNNs profundas no ha sido plenamente explorado en imágenes ortogonales y casi-ortogonales de alta resolución, especialmente en la conservación de especies de plantas.

Esta tesis presenta uno de los primeros estudios en explorar el potencial de las CNNs profundas, el preprocesamiento de datos y los datos de teledetección, para abordar los problemas de conservación de especies de plantas. En particular, esta tesis presenta los resultados y el análisis de modelos CNN profundos en tres problemas diferentes de ciencias naturales:

1. La detección de abetos (*Abies Sibirica*) dañados por el escarabajo de la corteza en imágenes de UAV utilizando DL.

2. La estimación del biovolumen de los olivos a partir de la segmentación de imágenes de UAV de múltiples resoluciones utilizando Mask R-CNN.

3. La detección de abetos (*Picea Abies*) infectados por el escarabajo de la corteza en imágenes UAV utilizando arquitecturas YOLO.

El objetivo principal de esta tesis es desarrollar modelos de aprendizaje profundo robustos y precisos para la monitorización de diferentes especies de plantas utilizando imágenes de UAV. Los objetivos particulares para lograr el objetivo principal son:

- Construir tres conjuntos de datos de alta calidad para cada uno de los tres problemas considerados según la formulación del problema.

- Diseñar los métodos de preprocesamiento adecuados que reduzcan el ruido y la incertidumbre en las características.

- Desarrollar modelos robustos y precisos basados en CNN para cada caso de estudio.

Los resultados de los primeros capítulos de de esta tesis han sido publicados en dos revistas clasificadas como Q1 y Q2 en JCR. Los resultados del tercer capítulo han sido sometidos a la revista "IEEE Transactions on Geoscience and Remote Sensing".

Esta tesis está estructurada en cinco capítulos. El primer capítulo presenta una introducción de los problemas considerados, los antecedentes y los objetivos de la tesis. El capítulo dos presenta el conjunto de datos construido, los modelos y la técnica de preprocesamiento para abordar la detección de abetos (*Abies Sibirica*) dañados por el escarabajo de la corteza en imágenes de UAV. El capítulo tres presenta el conjunto de datos construido, los modelos y la técnica de preprocesamiento para estimar el biovolumen de los olivos a partir de la segmentación de imágenes multirresolución de UAV. El capítulo cuatro presenta el conjunto de datos construido, los modelos y la técnica de preprocesamiento para abordar la detección de abetos (*Picea Abies*) infectados por el escarabajo de la corteza en imágenes UAV utilizando arquitecturas YOLO. Por último, el capítulo cinco presenta las conclusiones y el trabajo futuro.

I. Introduction

Deep Learning (DL) models, also called deep neural networks (DNNs), have shown outstanding performance in several fields of application, especially in computer vision and natural language processing. In particular, Convolutional Neural Networks (CNN), a special type of NN, constitutes the state-of-the-art in all computer vision fundamental tasks, namely image classification [KrSH12], object detection [RHGS16], and segmentation [HGDG18]. The first implementation of CNNs was done by Kunihiko Fukushima in 1983 [FuMI83]. Its structure was inspired by the visual cortex [MMMK03]. In 1988, Yann LeCun [LBDH89] showed that when trained on a large dataset such as MNIST, a simple CNN architecture, such as LeNet, can achieve impressive results of 0.8% error. However, only in the mid-2000s, CNN began to gain popularity thanks to the advancement of NVIDIA GPUs technology, which substantially accelerated the learning process, and also thanks to the first massive dataset named ImageNet. This made it possible to create a more complex NN architecture with a performance that exceeds human efficiency [CiMS12].

Remote sensing (RS) data is extremely useful for Earth observation, and it is a powerful tool for tracking a wide range of phenomena. Much has changed since the first image of the earth that was taken in 1946 [Reic06]. Now, RS is considered as one of the fastest growing areas of modern technology. High prospects are associated with the use of unmanned aerial vehicles (UAV), lidars, and microsatellites. The relevance of RS is increasing with the advent of new sensors and platforms for obtaining images with high and ultra-high resolution, less than 1 meter per pixel. Obtaining RS data is becoming less costly and more accessible for more users.

The combination of DL with RS data provides important opportunities for addressing new real-world problems, especially in the field of natural sciences. Hereby, the proposed thesis is exploring the potential of DL in the monitoring and detection of plant species using high resolution RS data. More specifically, this thesis addresses three problems:

- The detection of Fir trees (*Abies Sibirica*) damaged by the bark beetle in UAV images using DL. This problem will be analyzed in Chapter II.
- The estimation of olive tree biovolume from UAV multi-resolution image segmentation using Mask R-CNN. This problem will be analyzed in Chapter III.
- The detection of Spruce trees (*Picea Abies*) infected by bark beetle in UAV images using YOLOs architectures. This problem will be analyzed in Chapter IV.

This Chapter I provides a brief description of the main concepts used in this thesis. The main objectives of the thesis are highlighted in Chapters II-IV, the conclusion is given in Chapter V.

1. Background

This section presents the main concepts that have driven this thesis. Section 1.2 describes RS images and preprocessing. Section 1.2 introduces DL models for image processing, namely image classification models in Subsection 1.2.1, Object detection models in Subsection 1.2.2, and image segmentation models in Subsection 1.2.3.

1.1. RS Images and Preprocessing

RS is the process of obtaining information about an object or phenomenon without making physical contact with that object on the Earth. This process is carried out by measuring the reflected or self-radiation from a certain distance using either a passive or active sensor [Scho06]. The result of this process is an image of the territory, which displays real information at a certain point in time. RS data can be divided into two different types depending on the means of its acquisition: aerial photography and satellite imagery.

Aerial photography are images taken from an aircraft [Scho06]. Aerial vehicles used for aerial photography include fixed-wing aircraft, helicopters, unmanned aerial vehicles (UAVs), balloons, blimps and dirigibles, rockets, pigeons, kites, parachutes, stand-alone telescoping, and vehicle-mounted poles. Aerial photography can be divided into (a) black and white images, (b) color images, consisting of three spectral channels as blue, red, and green (RGB), (c) infrared images, consisting of RGB channels and infrared channels [PaKi12].

Satellite imagery constitutes the images taken by Earth satellites. Satellite imagery is divided into (a) color images, (b) infrared images, (c) multispectral images, consisting of 7-12 data channels, (d) hyperspectral images, consisting of up to 50 or more channels.

The most relevant characteristic of RS images is the spatial resolution which determines the pixel size of an image, as it is described in Table 1.1.

Table 1.1. The characteristic of RS images is the spatial resolution which determines the pixel size.

Spatial resolution types	Resolution (m/pixel)
Very low resolution	less than 100
Low resolution	15-100
Average resolution	5-15
High resolution	1–2.5
Ultra-high resolution	0.3-1

Currently, the most popular method for local Earth sensing is the UAV with digital, quite often semi-professional, devices. UAV images reach ultra-high resolution of up to 3 cm/pixel. The advantages of such a survey approach include its ability for urgent monitoring of the Earth's surface in small areas within a relatively short period of time (even under the condition of low continuous clouds), the relatively low weight and ease in the management of the system, as well as lower economic costs.

UAV images must be preprocessed before being delivered to users. This preprocessing includes data cleaning and data optimization.

Data cleaning is carried out in order to exclude various kinds of factors that reduce the quality of the data and interfere with the work of analytical algorithms. It includes processing of duplicates, inconsistencies and dummy values, restoration and filling of gaps, anti-aliasing, noise suppression and editing of anomalous values. In addition, during the cleaning process, violations of the structure, completeness and integrity of data are restored, and incorrect formats are converted.

Data optimization as a preprocessing element includes dimensionality reduction, identification, and elimination of insignificant features.

The main difference between optimization and cleaning is that the factors eliminated during the cleaning process significantly reduce the accuracy of solving the problem or make the work of analytical algorithms impossible. Optimization problems adapt the data to a specific task and increase the efficiency of their analysis.

Preparing RS data can include several steps, such as: (i) the transformation to an orthoimage, (ii) contrast enhancement of the image, if necessary, (iii) calculation of vegetation indices if the spectral channels are available and in accordance with the task at hand, (iv) selection of research objects, and (v) annotation of selected features based on ground measurements.

1.2. DL Models for Image Processing

Currently, Deep CNNs are the most accurate methods for extracting spatial patterns in analyzed data. Actually, CNNs constitute the state-of-the art models in all fundamental tasks in computer vision, image classification [KrSH12], object detection [RHGS16], and segmentation [HGDG18]. Classification models are the basis for all the rest of tasks. That is, to build good detection and segmentation models it is essential to use a high-performing classification model. This section provides a brief description of these fundamental models.

1.2.1. Image Classification Models

In image classification, the CNN analyzes a given input image and outputs a label that describes the object-class presented in that image. The most influential CNN architectures in the task of image classification are:

AlexNet is a DNN created by three scientists from the University of Toronto. This architecture won the ImageNet classification competition. The network architecture contains eight weighted layers [KrSH12]. The first five layers are convolutional, and the rest are fully connected layers. Based on the results obtained, this CNN is capable of achieving record results on very complex datasets (1 million photos), using only supervised learning and giving an average error of 15% from the 2012 course results.

Visual Geometry Group (VGG) is a NN architecture that has four variants, VGG-11, VGG-13, VGG-16 and VGG-19, where the number 13, 16 and 19 indicate the number of layers in the network. VGG-16 was designed by the University of Oxford to recognize objects in images. VGG-16 network is the 1st runner-up on the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) comparison in 2014 with an obtained accuracy of 93.3% [SiZi15]. A distinctive feature of the architecture is a small convolution kernel of 3 × 3 pixels. The NN architecture consists of two parts. The first part consists of alternating convolution cascades and a pooling layer: two convolution-convolution-pooling cascades and then three convolution-convolution-convolution-pooling cascades. On the pooling layer, Max

Pooling is selected with a 2 × 2 square core. This part highlights the characteristic features in the image. The second part is responsible for the classification of the object in the image according to the features selected at the previous stage and includes three fully connected layers. Thus, the VGG-16 network receives images with a size of 224 × 224 pixels in three color channels (RGB) and the output represents the probability of belonging to a particular class in one hot encoding format.

ResNet was developed by Microsoft using residual learning for recognition, localization and detection of objects in images. It was the winner of the ILSVRC 2015 competition with 150 layers and the winner of Microsoft Common Objects in Context 2015 (MS COCO) detection and segmentation [HZRS15]. The authors of ResNet used the two-layer traversal approach and applied it on a large scale, which is considered as a small classifier in the network. The network also uses a bottleneck layer, which allows the reduction of the number of features in each layer, using a 1 × 1 pixel convolution with a lower yield of features and then a 3 × 3 pixel convolution layer and again a 1 × 1 pixel convolution layer with more features. The bottleneck layer reduces computational resources while preserving feature combinations. The output layer is the pooling layer with the Softmax function.

Inception-V3 is Google's CNN for recognizing objects in an image. In December 2015, the third version of Inception was introduced with the inclusion of Batch-normalized Inception. Batch-normalization calculates the mean and standard deviation of all feature distribution maps in the output layer and then normalizes their responses with these values [SVIS15]. The last layer of the Inception network is the pooling layer with the Softmax function.

InceptionResNet-V2 is the union of two NNs: Inception and ResNet. This allowed the authors to improve the accuracy of image classification. The idea of residual blocks was preserved from the Inception network and a combination of convolutional blocks from the ResNet network [SIVA16]. Despite the fact that InceptionResNet-V2 demonstrates high accuracy compared to other existing models, it has two drawbacks: low speed and a high amount of memory used.

Xception was implemented in Keras by François Chollet and it is a modification of the Inception network [Chol17]. The network architecture is similar to ResNet-34 but the model and code is simpler than in Inception. A distinctive feature of the network architecture is Separable Conv, which is a modifiable, separable convolution that is located at the top of the network architecture. In the network there are residual (or shortcut/skip) connections, taken from the network ResNet. In addition, the network contains residual connections, which significantly increase the accuracy of classifying objects in images.

DenseNet is the result of the development of the ResNet network and is based on its residual blocks [HLMW18]. The basic idea is that connections have all the possible combinations within each block, which represents a gradient of more paths, and the network becomes more resistant to learning. There are few types of networks: DenseNet-121, DenseNet-169 and DenseNet-201.

1.2.2. Object Detection Models

In object detection, the CNN analyzes the input image and outputs a label together with a bounding box that delimits where the object-class is located in the image. In general, object detection models combine a search algorithm with a classification model. The search algorithm tries to generate areas of the image that are more likely to contain an object, for example, the sliding window technique.

Modern detection models can be divided into two categories, two-stages detection models and one-stage detection models. Two-stage models are region proposal based models that operate in two stages, first they search for candidate regions then in a second stage they analyze all these regions using a CNN feature extraction model. Examples of this class are Region-Based Convolutional Neural Network (R-CCN) [GDDM14] and all its variants Fast R-CNN [Girs15], Faster R-CNN [RHGS16]. Whereas one-stage detection models perform the search and feature extraction at the same time. Examples of these models are YOLO and its variants.

Two-stage detection models: The first detector in this category is R-CNN. It was developed by Ross Girshik et al. [GDDM14] in 2014. This model is based on the following steps:

1. First it uses the Selective Search method [USGS13] to find potential objects in the input image and outputs region proposals.

2. It analyzes each region using a CNN-based features extractor.

3. It classifies features using the Support Vector Machine method [PiSc20].

4. Finally, it applies a refinement of boundaries of the obtained bounding boxes using linear regression.

As a result, from using R-CNN, a separate region with an object and its class is displayed. However, the proposed method is time-consuming and requires a lot of training time and cannot be used for video sequences. In 2015, the same authors improved the model and named it ***Fast R-CNN***. It is based on the following architecture:

The image is fed into the input of a CNN and processed by the Selective Search method. As a result, at this step, a feature map and regions of potential objects are formed. The coordinates of the regions of potential objects are converted to coordinates on the feature map.

The resulting feature map with regions is passed to the Region of Interest (RoI) pooling layer. Here, an H × W grid is superimposed on each region. Then MaxPolling is applied to reduce the dimension of the image. So, all regions of potential objects have the same fixed dimension.

At the next step, the obtained features are fed to the input of a Fully connected layer, which is passed to two other fully connected layers. The first, with the Softmax activation function, determines the probability of belonging to the class, the second - the boundaries (offset) of the region of the potential object.

Thus, Fast R-CNN demonstrated higher accuracy and greater increase in processing time, in contrast to the first version. However, this method still requires computing power due to Selective Search, which led the authors to improve the method to ***Faster R-CNN***.

In 2016, the authors proposed a new method to localize the Region Proposal Networks (RPN) object instead of the Selective Search method [HZRS15]. The method is based on a system of anchors.

The Faster R-CNN architecture has been improved as follows:

The image is fed to the input of a CNN and then a feature map is formed.

The feature map is processed by the RPN layer. Here the sliding window is traversed over the feature map. The center of the sliding window is linked to the center of the anchors. Anchors are areas that have different aspect ratios and different sizes. Based on the intersection-over-union metric, the degree of intersection of anchors and true marked rectangles, a decision is made about the current region is there an object or not.

Next, the Fast R-CNN algorithm is used: the feature map with the obtained objects is transferred to the RoI layer, followed by processing of fully connected layers and classification, as well as determining the displacement of the regions of potential objects.

Consequently, the Faster R-CNN model is faster, but gives slightly worse localization results than Fast R-CNN [Girs15].

One-stage detection models: this category of detector performs the search process and feature extraction at the same time. They are generally faster than two-stage detectors. One of the main examples of this category is YOLO, which is a real-time CNN for object detection. This model was first proposed by Joseph Redmon in 2016 [RDGF16]. The model is capable of detecting 20 different classes. Currently, there are 4 different modifications of the YOLO model in the literature. The second version of ***YOLOv2*** was proposed by Joseph Redmon and Ali Farhadi in 2016 [ReFa16], its new modification ***YOLOv3*** was presented by the same authors in 2018 [ReFa18], and the last updated version ***YOLOv4*** was proposed by Alexey Bochkovskiy in 2020 and 2021 [BoWL20, WaBL21].

Most object detection algorithms take in and process the image multiple times to be able to detect all the objects present in the images. But YOLO looks at the object only once (hence the name – You Only Look Once). It applies a single forward pass to the whole image and predicts the bounding boxes and their class probabilities. The architecture consists of two major components: feature extractor and feature detector or multi-scale detector. The image is first given to the feature extractor which extracts feature embeddings and then is passed on to the feature detector part of the network that produce the processed image with bounding boxes around the detected classes.

If we compare the selected architectures, then YOLOv1 with the Darknet-19 NN results in more localization errors but is much less likely to predict false positives when searched objects do not present in the data. It outperforms all other detection methods, including deformable part models (DPM) and Regions-based CNN (R-CNN). However, despite the improvements in the YOLOv2 version on Darknet-30 networks, it has better results of detection, but also has a problem while detecting small objects due to down sampling the input image and losing fine-grained features. The improved architecture of YOLOv3 with Darknet-53 and ResNet networks due to its complexity is a bit slower compared to YOLOv2, but at the same time, gives results with higher accuracy. The improved YOLOv4 architecture, in contrast to the previous version, works much faster without loss of definition quality.

1.2.3. Image Segmentation Models

In image segmentation, the CNN analyzes the input image and outputs a label together with a polygon that delimits the pixels of each instance of the object-class. One of the most widely used methods is recently developed ***Mask R-CNN*** [HGDG18]. This method extends Faster R-CNN to solve segmentation tasks. It achieves this by adding a branch for predicting an object mask in parallel with the existing branch for bounding box recognition. Thus, Mask R-CNN is designed for pixel-to-pixel alignment between network inputs and outputs. Mask R-CNN analyzes an input image and provides three outputs for each object-class: (1) a class label that indicates the name of the object-class, (2) a bounding box that delimits each object-class and (3) a mask that delimits the pixels that constitute each object-class.

Mask R-CNN consists of the following structural components:

- The base, that is the standard ResNet50 CNN, at the early layers of which objects of a low level (edges and corners) are detected, and at the later layers of the network elements of a higher level (tree, person, building) are found. Going through the NN, the image is converted from an m×n×3 array, where m × n is a matrix of pixels and 3 is an RGB channel, to a 32×32×1024 object map. This feature map becomes the entry for the next network levels. To improve the quality of extraction of objects by the network, we used the feature pyramid network [LDGH17], which takes the high-level elements detected by the CNN and transmits them down to the lower layers. This allows functions at each level to have access to functions at both the lower and higher levels.
- Region Proposal Network. This is a CNN, which scans an image in a sliding window mode and finds candidate areas containing objects. As a result, the network generates an anchor class (foreground and background, where the first implies the presence of the object being classified), and a bounding box with the delta estimate (percentage of the relative measurement accuracy of the coordinates, width, and height) for specifying the anchor field to better match the object.
- Classifier and bounding regressor window. This component generates two outputs for each object: object class and bounding box. This frame is necessary to further clarify the location and size of the object.
- Segment mask, that is a CNN that takes as input the areas of objects selected by the classifier and generates masks for them. The generated masks have a low resolution of 28×28 pixels. The small size of the mask helps keep the mask network light. During training, we reduce the masks to 28×28 pixels to calculate the losses, and during the output we increase the predicted masks to the size of the bounding box of the object, and they are final masks, one per object.

2. Objectives

After introducing the main concepts related to this work, we present the main objectives that have driven this thesis. They include the study and analysis of different DL based approaches to address real world RS problems concerned with plant species monitoring and detection. Specifically, the final objective of this thesis is to determine how DL and RS data can be used together to solve the three following problems from natural sciences:

- The detection of fir trees (*Abies Sibirica*) damaged by the bark beetle in UAV images with DL.
- The estimation of plant biovolume based on UAV images and multi-resolution image segmentation with Mask R-CNN
- The detection of spruce trees (*Picea Abies*) infected by Bark Beetle in UAV images using YOLOs architectures.

It is worth mentioning that these problems were addressed for the first time in the context of this thesis.

3. Methodology

To achieve the objectives in each problem of the three considered case studies, we followed the classic scientific method illustrated in Figure 1.1. After defining the problem, the used methodology consists of formulating the hypothesis, data acquisition, design and development of the algorithm, evaluation of the overall performance, and finally adapting the initial hypothesis depending on the obtained results. This process will be applied iteratively until we improve the quality of the data and models to the level required for the successful solution of the considered problem.

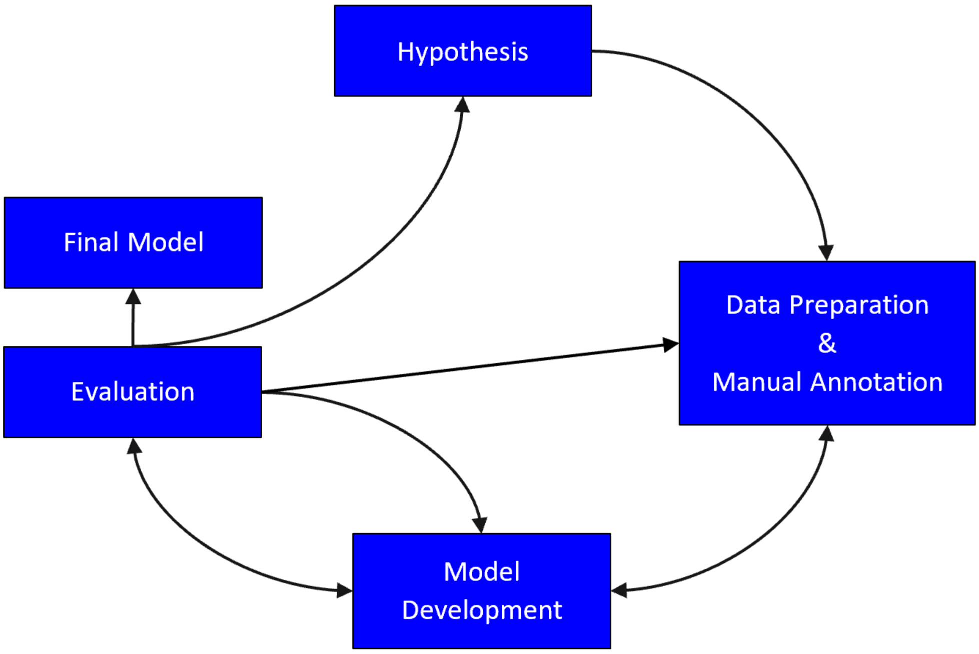


Figure 1.1. The diagram of the scientific methodology followed for the development of the three biodiversity problems.

The reformulation of the hypothesis consists of expressing the solution as either a classification, detection, or segmentation model, or a combination of multiple models. The most critical stage in this process is data preparation as it requires building good quality training datasets to be able to obtain high-performing models. We will use public repositories and/or private resources.

II. Detection of Fir Trees (*Abies Sibirica*) Damaged by the Bark Beetle in Unmanned Aerial Vehicle Images with DL

1. Introduction

Taiga and Boreal forests play an important role in the global climate through the carbon, water and energy balance [Bona08]. In Russia, fir forests provide multiple provisioning, regulating and cultural ecosystem services and are considered a national asset of the country. Despite forests are a renewable resource, forest degradation is too high and not compensated by regeneration [HPMH13]. One of the main problems of Russian forest degradation is the spatial spread acceleration of the Four-eyed fir bark beetle (*Polygraphus proximus* Blandford) invasion that causes fast death of fir trees (*Abies Sibirica* Ledeb) in various forest ecosystems. The geographic origin of *P. proximus* propagation comes from its natural habitat in Japan, the Korean Peninsula, Eastern China, and the Russian Far East (Khabarovsk and Primorskii krai, Sakhalin, and Kuril Islands) [Леp97]. In Siberia, the first beetle occurrence was registered in 2008 [Kerc14, BaAA11] and within the last 10 years, massive outbreaks of *P. proximus* occurred on large forested areas in the southeastern part of the West Siberian Plain: Tomsk, Kemerovo and Novosibirsk oblasts, Altai region, in the Altai Republic as well as in Krasnoyarsk region [BaAA11, PKUB18], where the problem became catastrophic and got out of control. The invasion of this type of beetle can lead not only to degradation of fir forests, but also to create a threat to fir existence as a forest type species, with the subsequent broad implications for the regional and global climate [HWKC16, Ma16].

Death of fir trees due to *P. proximus* outbreaks occurs in several stages. The beetle usually attacks trunks of weakened trees, fallen deadwood, and newly harvested wood. In case of massive outbreaks *P. proximus* also attacks healthy trees, which can resist against the beetle attack during 2-3 years. Penetration of *P. proximus* under tree bark results in pervasion and reproduction of ophiostomatoid fungi (different species), their phytopathogenic activity leads to a gradual weakening of fir trees. Further the beetle colonizes a tree which begins to dry out. Increase of beetle quantity in a local forest stand leads to a massive death of fir trees. Usually, fir trees die within 2-4 years from the moment of first beetle attack, so the task of the beetle invasion monitoring is really important and poses a challenge to develop a remote survey system for precise estimation of forest damage states.

Advances in Earth remote sensing (ERS) techniques, particularly very high resolution satellite and airborne imagery, open the possibility to develop tools for regional mapping of consequences of forest pest activities. Nowadays, application of unmanned aerial vehicles (UAV) tends to be more popular in local scale forestry research because of better spatial resolution [LNPK15]. Also, UAV data provides the basis for development of new methods in data analysis which can be applied later at larger scales (satellite data). Recently, application of DL, in particular CNNs, in processing of color (RGB) images of the Earth surface from various data sources have provided high accuracy results in recognition of different plant species [BNMA17, KLSS17, LFYC16, Remo00, WKJS14]. However, there are still no published studies on the application of CNN methods to very high resolution imagery for detection of forest health decline caused by pest invasions. The goal of this study was to test the possibilities of NNs as a new approach to detect bark beetle outbreaks in fir forests. In particular, our aim was to develop and test a CNN method to automatically detect individual fir trees disturbed by *P. proximus* in very high resolution imagery of mixed forests.

The main contributions of this chapter can be listed as follows:

- As far as we know, this is the first work in addressing the problem of forest damage detection caused by the *P. proximus* beetle in very high resolution images from UAVs with DL considering four tree categories (classes).
- We built a new labeled orthoimages data set of four fir trees' damage stages.
- We designed a new CNN architecture to accurately classify trees in UAV images into different damage categories of trees.
- We developed the detection model as follows. First, a new detection method selects the candidate regions that contain trees in UAV images. Then, these candidate regions are processed by the multiclass model to finally predict the category of tree damage.
- We provide a complete description of the used methodology and the source code so that it can be replicated by other researchers to detect and classify tree damage by bark beetles in other images. Our source code can be found at https://github.com/ansafo/OurCNN.

The rest of this chapter is organized as follows. The tree classification and related works are given in Section 2. The study area and data acquisition are presented in Section 3. The proposed methodology is presented in Section 4, which includes a definition of fir trees damage categories, preprocessing dataset and data augmentation techniques of sample patches for training CNN models, description of the development of the classification model algorithm and creation of a data subset for independent model verification. Section 5 presents the obtained results and data analysis. Discussion and conclusions are given in Section 6.

2. Classification of trees in high resolution imagery and related works

In this section, we consider related works devoted to the problems of classification and detection of objects on ERS data. Scientists from around the world are trying to solve this problem using various ERS information with different methods and algorithms.

For example, Deli et al. [DeBY16] developed a new CNN-based method for classifying four land-cover classes (crops, houses, soil and forests) in Earth surface images. They trained the algorithm with 100 images per class. Längkvist et al. [LKAL16] proposed the classification and segmentation of satellite orthoimagery (included five classes: vegetation, ground, road, parking, railroad, building and water) using CNNs in a small city for a full, fast, and accurate per-pixel classification. They selected the parameters and analyzed their influence on the NN model architecture. A comparison of the CNN model with object-oriented methods of classification is presented, where the maximum CNN classification accuracy is 94.49%.

Few works use NN classifiers to recognize plant species and plant growth using images from a digital camera. Dyrmann et al. [DyKM16] created a CNN capable of recognizing 22 plant species on color images with an accuracy of 86.2%. To solve the problem, the authors used different data sets, depending on the lighting, image resolution, and soil type. The images were captured using a digital camera and a cell phone camera. Razavi and Yalcin [RaYa17] proposed a CNN architecture to classify types of plants growing in smart agro stations. To evaluate the effectiveness of the approach, the results of the created CNN model are presented in comparison with the results obtained using the SVM (RBF kernel) and SVM (polynomial kernel) classifiers. The accuracy of the CNN classification was 97.47%.

The next series of works using CNN represent the high results of different tasks using a variety of satellite imagery. Thus Li et al. use DL models in [LFYC16] to detect oil palm trees on QiuckBird images (0.6 m/pix). Guirado et al. in [GTAC17] detect shrubs *Ziziphus lotus* on satellite images from Google Earth in European Petroleum Survey Group (EPSG) with resolution as up to 0.12 m/pix. Baeta et al. in [BNMA17], the authors recognize the coffee crop on high-resolution SPOT images (2.5 m/pix). They used a binary classification model and presented a comparison of the accuracy of various CNN and object-based image analysis (OBIA) methods. In addition, the authors use a large number of methods of preparation and preliminary processing of the image, which in the final result leads to a high accuracy of more than 95%.

Also noteworthy is the work where [WFSH18] weed was detected in soybean cultures using CNN based on the CaffeNet architecture. The authors used a set of data acquired with an UAV in manual mode at a height of 4 meters above the Earth level using an RGB camera. The accuracy of the classification was 99.5%. Though, according to the authors, in practice the accuracy may be lower due to various types of soil and weeds, since testing was performed on an experimental field.

A similar problem to ours, which is the detection of damaged trees, was considered in the next series of works. In 2009, Groen et al. [ADSG18] identified healthy and infested trees in satellite imagery (Sentinel-2, WorldView-2 and 3 (up to 0.5 m/pix)) using partial least squares regression (PLSR). Heurich et al. [HOAS13] presented semi-automatic detection of dead trees following a spruce bark beetle on CIR aerial photographs using object-oriented image analysis with accuracy up to 91%. In this work authors used several classes, as deadwood areas, vital vegetation, non-woodland, and assistant classes (shadows among dead trees and shadows among vital trees). In works from 2013 Ortiz et al. [OrBK13] detected of Bark Beetle Green Attack with TerraSAR-X and RapidEye Data (up to 1.18 m/pix) used generalized linear models (GLM), maximum entropy (ME) and random forest (RF) up to 74%. Meddens et al. [MHVH13] used multi-temporal disturbance detection methods to detect bark beetle-caused tree mortality and mapping of forest disturbances on Landsat images with different classes (green trees, red stage, gray stage, herbaceous vegetation, bare soil, shadow/water). Kussul et al. [KLSS17] classified tree species and different levels of ash mortality using classical methods, in particular, the linear discriminant analysis (LDA), principal component analysis (PCA), a stepwise selection method, OBIA methods were used on WorldView-2 image data with resolution of 1.85 m/pix. Species diversity and the magnitude of ash (Fraxinus sp.) stand loss were classified. The authors used a set of remote sensing indices (RSI) to obtain the best result. The overall accuracy varies between 83% for the seven tree species and 77% for the four different levels of damaged ash. From 2015 Näsi et al. [NHLB15] identified damaged trees (with three classes: healthy, infected and dead) UAV images using method based object based. In 2017, Dash et al. [DWPH17] demonstrated monitoring forest health for disease outbreaks in mature Pinus radiata D. Don trees in the task of classification trees on UAV images using a non-parametric approach. Onishi et al. [OnIs18] classified 7 types of trees (deciduous broad-leaved tree, deciduous coniferous tree, evergreen broad-leaved tree, Chamaecyparis obtusa, Pinus, Pinus Strobus, others) used supervised DL (GoogLeNet) with an accuracy of 89% on UAV images. Also in this year, a paper was published that most closely matches ours, where Näsi et al. [NHLB15] identified of bark beetle infestations at the individual tree level (healthy, infested, and dead) in urban forests on UAV images used support vector machine (SVM) an accuracy of 93%.

In general, most previous works in the detection of damaged fir trees using UAV images consider few classes, healthy and dead trees, and did not apply CNNs.

As far as we know, this work is the first in addressing the detection of damaged fir trees caused by the bark beetle using CNN methods on images acquired by UAVs (resolution below 0.01 m/pix). We considered four categories, which is a higher number of categories than previous works. We also showed that it is possible to achieve good results using a relatively small training set of data.

3. Study area and data acquisition

The study area is located in the territory of the state nature reserve “Stolby” which is situated near the Krasnoyarsk city in Central Siberia of the Russian Federation. Most of the territory (80%) constitutes the mid-mountain belt (500 - 800 m a.s.l.), mainly covered by mixed forests composed by seven tree species in different proportions: conifers such as pine (*Pínus sylvéstris*, *Pínus sibírica*), larch (*Lárix sibírica*), fir (*Ábies sibírica*), spruce (*Pícea ábies, Pícea obováta*), and perfoliate forest such as birch (*Bétula pendula*, *Bétula pubéscens*) and aspen (*Pópulus trémula*) [Ряб013]. Pine trees dominate among other species and occupy 41% of the total forested area mainly in low mountains. Distribution of Siberian fir expands to 25% of the Stolby forest.

First appearance of *P. proximus* in the Stolby nature reserve was registered in 2011, and nowadays the roughly estimated forest damage by the beetle is about 25-30% of the fir area.

Four plots with different rates of the *P. proximus* invasion were chosen for our study at Stolby (Figure 2.1).

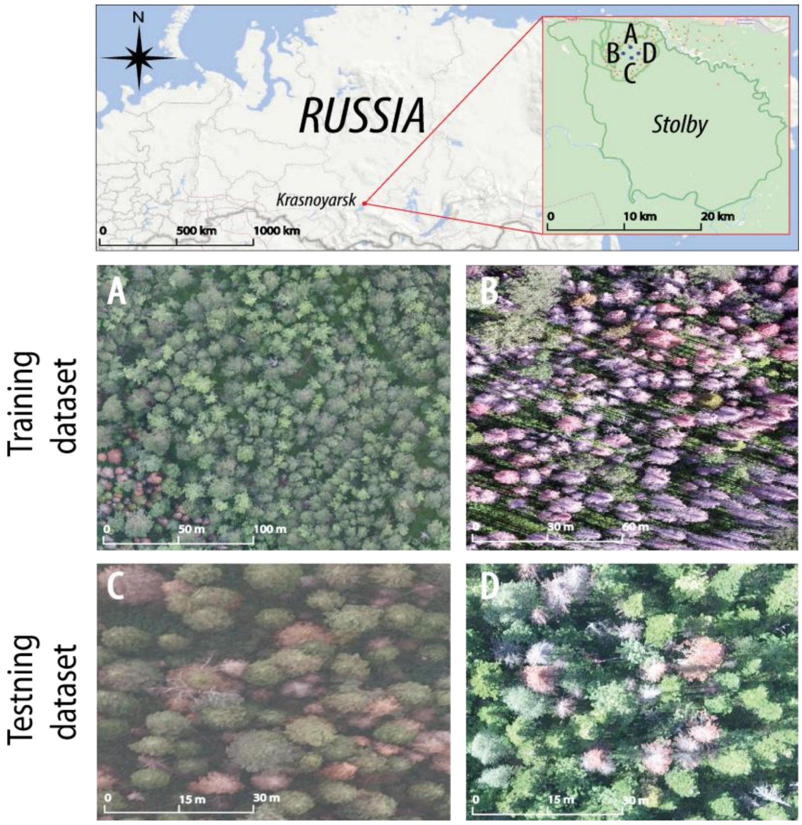


Figure 2.1. Location of the four plots in the nature reserve "Stolby", Krasnoyarsk city (Russia), where **A** and **B** plots are fragments from the orthophotos used to build the training dataset; **C** and **D** plots are fragments from the orthophotos used to build the testing dataset used for external validation or independent testing.

A set of RGB images with ultra-high spatial resolution (≈5-10 cm/pix) were obtained for the research sites during multiple flights of the DJI Phantom 3 Pro quadcopter (with standard camera) in July 2016 (plot A and C) and the Yuneec Typhoon H hexacopter (with CGO3+ camera) in May 2016 (plot B) and August 2018 (plot D). Imagery for research plots A and C were recorded in cloudy weather conditions at 670 m (A) and 700 m (C) altitudes (120-150 meters elevation above ground), plot B and D was surveyed in sunny weather at 120 m height. Default camera settings (auto white balance, ISO 100) were applied in all aerial shots. Image composites (orthophotomosaic) were created from a set of multiple images (300-400 per plot) using the Agrisoft Photoscan software. As a result, we got four orthophotos (2016 and 2018) for each one of the four sites (A, B, C, D).

4. Methods

As we have mentioned, our aim is to evaluate the possibilities of NNs as a new approach to detect bark beetle outbreaks in fir forests in very high resolution imagery of mixed forests. Thus, this section includes a definition of fir trees damage categories (subsection 4.1), describes the data preprocessing techniques we used for training the CNN models (subsection 4.2), presents the classification model (subsection 4.3) and finally presents the proposed detection technique especially designed for detecting bark beetle outbreaks in fir forests (subsection 4.4).

4.1. Definition of fir trees damage categories

To define the categories of health status of fir trees, we followed the entomological approach by Krivets and Baranchikov [PKUB18], which differentiates six categories according to the level *P. proximus* invasion into the trunk and its influence on the canopy: I – healthy trees; II – weakened trees; III – heavily weakened trees; IV – dying trees; V – recently died trees; VI – old deadwood. Since the first, second and third categories can only be recognized in-situ by trunk signs that do not visibly translate into the crown, we reclassified them into four categories (skipping the second and the third one). Hence, the final categories in our classification were: a – completely healthy tree or recently attacked by beetles, b – tree colonized by beetles, c – recently dead tree, and d – deadwood (Figure 2.2).

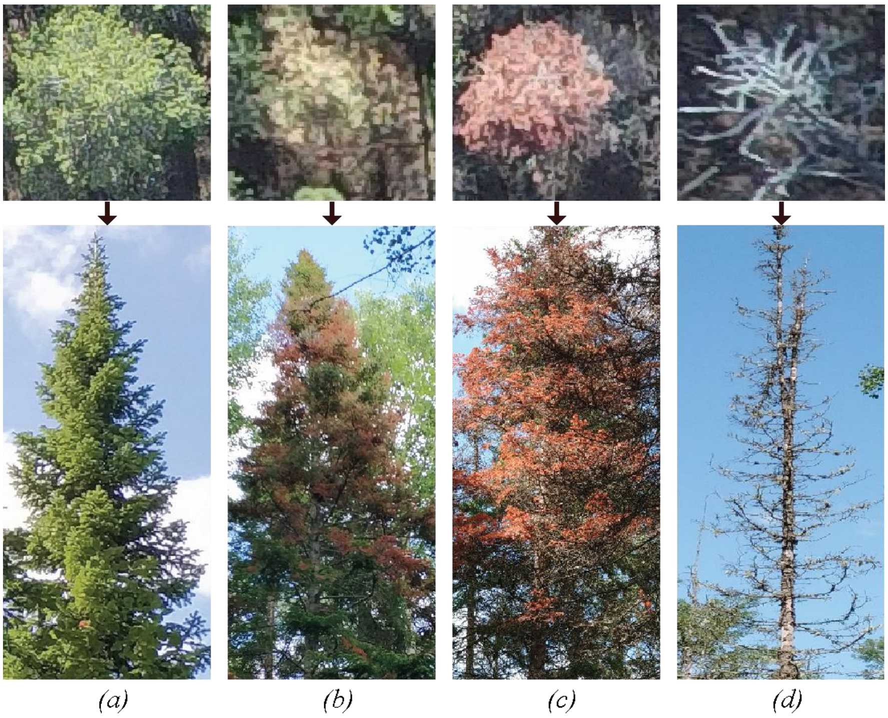


Figure 2.2. Damage categories of Siberian fir trees used in this study, where (**a**) completely healthy tree or recently attacked by beetles, (**b**) tree colonized by beetles, (**c**) recently died tree, (**d**) deadwood. Top figures illustrate the vertical orthoimages corresponding to the bottom horizontal pictures.

4.2. Dataset Preprocessing and data augmentation techniques of sample patches for training the CNN models

The design of the training dataset is the key to the performance of a good CNN classification model. For labeling the training dataset, we identified 50 150×200-pixels image patches containing 4 categories of trees. In particular, we built two different datasets:

- First, the dataset for training, validation and test consisted in 50 manually sampled image-patches of single trees per each tree damage category, resulting in 200 image-patches from the plot A and B. To train the CNN model we used 80% of this selection. The remaining 20% of images were used for model internal validation and test.

- Second, the dataset for external testing are 88 patch-images generated by our proposed detection technique from the areas C and D.

A summary of these two datasets is provided in Table 2.1.

The pre-processing method used for preparing the input UAV images is described in Table 2.2. To improve the robustness and accuracy of the CNN classification models, we increased the volume of samples using data augmentation techniques. In particular, we increased the amount of sample patches for training the CNN models using the methods.

Table 2.1. Four fir tree categories training and testing datasets. Each image has 150x200-pixels.

Categories of trees	Training dataset (plots A and B)				Testing dataset for external test	
	without data augmentation		with data augmentation			
	Training	Internal validation	Training	Internal validation	Test area C	Test area D
1	40	10	880	220	5	13
2	40	10	880	220	36	14
3	40	10	880	220	5	7
4	40	10	880	220	2	6
Total:	**160**	**40**	**3520**	**880**	**48**	**40**

Table 2.2. The stages followed for pre-processing the input UAV imagery and data augmentation methods used for increasing the amount of sample patches.

Steps	The pre-processing stages of input UAV imagery
Step 1	Conversion of the orthophotomosaics from .tiff to .jpg raster format.
Step 2	Selection of image patches – a set of pictures manually cropped from orthophotomosaics using QGIS 7.2.2.
Step 3	Resizing of patches to 150×200 RGB pixels using cubic interpolation.
Step 4	Manual assignment of each patch to appropriate tree damage category.
	Methods for increasing the amount of data
Step 1	Change the saturation of RGB channels.
Step 2	Remove the Gaussian blur filter with a blur value of 5% and a width and height of a kernel 0.5.
Step 3	Pixel averaging by collapsing an image with a normalized 4 × 4 window filter.
Step 4	Image rotation relative to its center with 5°, 15°, 50°, 90°, 180° rotation angles.
Step 5	Increase an image size from the center to 50%.

The images of the training dataset are used for training the parameters of the NN. The internal validation set is a set of examples used to tune the parameters of the NN and determine a stopping point for the back-propagation algorithm. The testing dataset is a set of examples used only to assess the performance of a final classification model.

4.3. Classification model

A deep NN was developed and trained on the prepared image sample to adapt its weights to the task of the tree's damage recognition. To create a CNN and improve the quality of its training we manually tuned the hyperparameters in the network. Series of experiments were performed, for each experiment the hyperparameters were altered and consequent network's operation quality change was estimated. When exploring the effect of changing the learning rate parameter with the Stochastic Gradient Descent optimizer for values 0.0001, 0.001, 0.01 and 0.1, the highest accuracy for CNN on the test data was achieved with a value of 0.0001. Creating cascades of convolution layers and a subsample layer, the total number of the network layers was determined. To assess the impact of the number of training epochs on the CNN accuracy, training was conducted in the range from 10 to 40 epochs.

The overall architecture of the CNN consisted of six convolutional blocks (each includes one convolutional layer). Additionally, the first and third convolutional blocks included maximum pooling layers. At the top of the CNN there are two fully connected layers and one output layer. The ReLU activation function was used in the last four convolutional blocks, and the Softmax function for the output layer. In order to keep the overtraining of the network under control, we used the Dropout regularization method which reduced the complexity of the model, saving the number of its parameters. It is very important to choose an appropriate regularization coefficient, so it was set to 0.25 after the second, forth, and fifth layers, and it was set to 0.5 before the output layer. Our network with various activation functions is presented in Table 2.3.

Table 2.3. Network with different activation functions, where layers, output size and network parameters are represented.

Layers	Output size, pix	Network
Convolution	150×200	3×3, 96
Max pooling	75×100	2×2, stride 2
Convolution	75×100	5×5, 128
Dropout	75×100	0.25
Convolution	75×100	3×3, 128
Max pooling	38×50	2×2, stride 2
Convolution	38×50	3×3, 128
Dropout	38×50	0.25
Convolution	38×50	5×5, 128
Dropout	38×50	0.5
Convolution	38×50	5×5, 512
Global Average pooling	1×972800	stride 1
Dense	1×972800	ReLu
Dropout	1×400	0.5
Dense	1×100	ReLu
Dense	1×4	Softmax

The objective function, which should be minimized in the NN, was established by the categorical cross-entropy loss between the input and the actual classification of images, which fits well to the output of the probability of the category's occurrence. The ADAM's optimization was chosen among existing optimization algorithms since it was a more efficient case due to the possibility of initial calibration of the CNN. Consequently, our CNN model has the following form, presented in Figure 2.3.

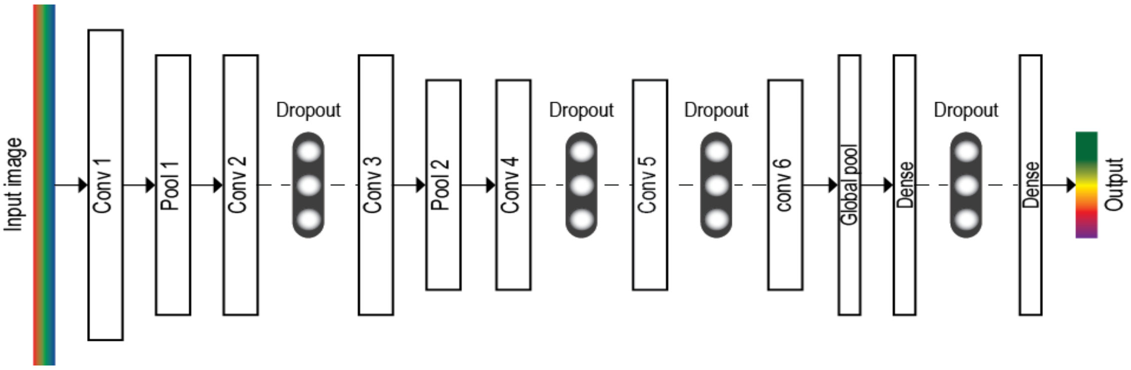


Figure 2.3. The architecture of our CNN model of deep machine learning.

4.4. Proposed detection process

We developed a candidate selection technique to find the potential regions in the test image that are more likely to be a tree. Then we analyze each one of these candidate regions using our CNN-classification model. The proposed candidate selection technique is a data processing algorithm that includes a sequence of the following steps presented in the Table 2.4:

Table 2.4. The proposed candidate selection technique in the test areas cropped from sections C and D.

Steps	Data processing algorithm
Step 1	Conversion of RGB images to a grayscale colors palette
Step 2	Blurring of grayscale images using the Gaussian high-pass filter in order to reduce Gaussian noise with the following function parameters: kernel size is 11x11 pixels, and the standard deviation equal to 0
Step 3	Creation of binary images from the grayscale blurred images by application of a threshold function with the optimal brightness threshold value of input image pixels equal to 100
Step 4	Structuring of picture elements outlines by application of two successive functions (erosion and dilation) with several iterations for the binary pictures in order to distinguish individual tree crown contours and to minimize the effect of their confluence (fusion) in one object
Step 5	Detection of image patches was implemented using a contour area calculation function based on the Green formula [Calc19], object size for the function was set in the range between 50x50 and 200x200 pixels

The output of this process are the candidate regions indicated by red bounding boxes in Figure 2.4. Finally, all the bounding boxes will be analyzed by the classification model.

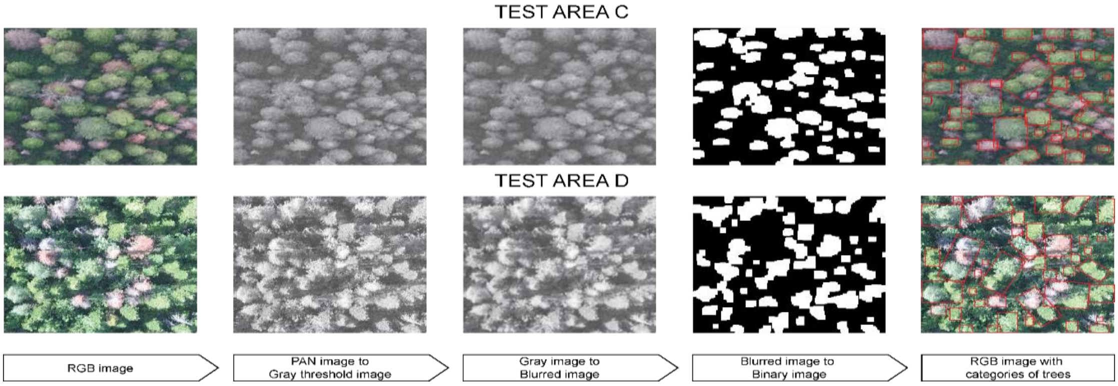

TEST AREA C
TEST AREA D
RGB image
PAN image to Gray threshold image
Gray image to Blurred image
Blurred image to Binary image
RGB image with categories of trees

Figure 2.4. Pre-processing consisted of the following steps: first, converting the three band image into one gray-scale band image (PAN), second, converting gray-scale band image into blurred image, third, converting the blurred image into a binary image based on a 100 over 256 digital value threshold, and then detecting categories of trees are show on RGB image. The 48 candidate patches identified in test area C and 40 candidate patches identified in test area D are labeled with red contour in the right panel.

4.5. Evaluation metrics

To evaluate and compare the results of the proposed CNN model with other CNN models, we used the performance metrics based on the confusion matrix. The confusion matrix is four by four (four categories) and contains the results of multi-class classifier (one class against the rest) in terms of, true positive (TP) prediction, true negative (TN) prediction, false positive (FP) prediction, and false negative (FN) prediction [MHVH13].

The main metrics we can extract from the matrix of confusion are accuracy (2.1), precision (2.2), recall (2.3), F-score (2.4). The accuracy, precision, recall and F-score. F-score indicates the balance between precision and recall. The highest and best value of all these metrics is 1.0 and the worst is 0.0.

$$Accuracy = \frac{TP+TN}{TP+TN+FN+FP} = \frac{TP+TN}{P+N}, \tag{2.1}$$

$$Precision = \frac{TP}{TP+FP}, \tag{2.2}$$

$$Recall = \frac{TP}{TP+FN}, \tag{2.3}$$

$$F\_score = 2 \times \frac{Precision \times Recall}{Precision+Recall}. \tag{2.4}$$

5. Experimental Results

In this section we will describe the performance of the training and internal test process and then we will show the results of the detection on two external test UAV images.

5.1. Evaluation of the training process and comparison

To evaluate the training process of our new NN, with different preprocessing data augmentation techniques, we used Python, Keras and TensorFlow framework. Keras is a high-level API to ease the process of building and training DL models [Kera21]. TensorFlow is an open-source software library for high performance numerical computation, which runs on various heterogeneous systems, including distributed graphics processing unit (GPU) cluster [Murr17]. The used hardware environment is Intel Xeon E5-2630v4 CPU accelerated with NVIDIA Titan Xp GPU as a platform for learning and testing.

The application of data augmentation techniques to the sample patches for training the CNN models resulted in the creation of 4400 image-patches from the first data subset, which has 200 patches (Section 4.2).

The maximum performance of our CNN model with augmented dataset was achieved at the 23rd training epoch providing an internal test accuracy of 99.7% and a minimum training loss less than 0.01. After the 23rd epoch, the validation loss was stabilized and the difference between the training and validation loss increased (Figure 2.5).

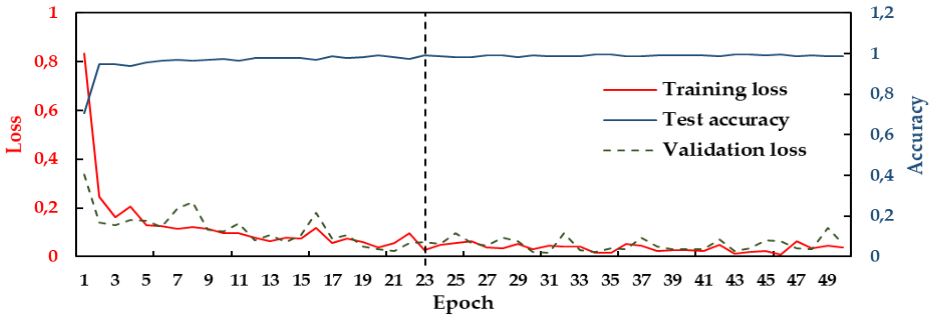


Figure 2.5. Loss and accuracy for each epoch of the CNN model training with data augmentation.

For comparison, we have tested the most powerful CNN models such as Xception, VGG16, VGG19, ResNet50, Inception V3, InceptionResNetV2, DenseNet121, DenseNet169 and DenseNet201 [Comm18] on the same input data. We adapted the last layer of all these models to the 4 classes of our problem. The obtained results of all these models after a few training epochs are shown in (Appendix A, Table A1). As it can be seen from Table A1, all models provide high training accuracies.

5.2. External test detection results

The confusion matrix of our CNN model in the four categories with data-augmentation are shown is Figure 2.6 and the results based on the confusion matrix of our CNN model on each category with and without data augmentation are shown in Table 2.5.

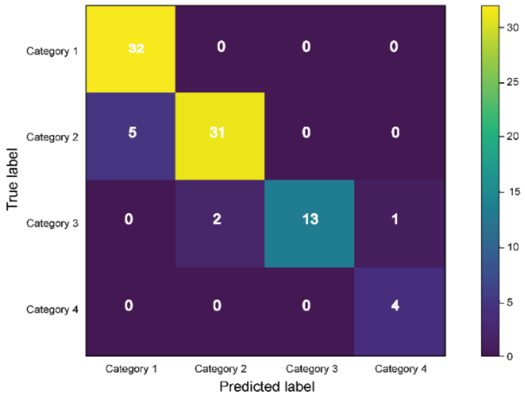


Figure 2.6. Confusion matrix of the proposed CNN model with data augmentation on the candidate regions obtained from test areas C and D.

Table 2.5. The performance of our CNN model with and without data augmentation on the test set for each category of trees. The performance is expressed in terms of true positives (TP), true negatives (TN), false positives (FP), false negatives (FN), precision, recall, and F-score.

Categories of trees	TP	TN	FP	FN	Acc (%)	Precision (%)	Recall (%)	F-score (%)
Without augmentation								
1	18	44	3	14	**78.48**	85.71	56.25	67.92
2	30	32	15	6	74.7	66.67	83.33	74.07
3	10	52	0	6	**91.18**	100	65.5	**76.92**
4	4	58	8	0	**88.57**	33.33	100	50
With augmentation								
1	32	48	5	0	**94.12**	86.49	100	**92.75**
2	31	49	2	5	**91.95**	93.94	86.11	**89.86**
3	13	67	0	3	**96.39**	100	81.25	**89.66**
4	4	76	1	0	**98.77**	80	100	**88.89**

As it can be seen from Table 2.5, the trained CNN with data-augmentation provides the best test accuracy, recall and F1_score. The average accuracy, recall and F-score improved by 14.5%, 20.4% and 34.3% respectively. A comparison between the results obtained by our model and by more complex models with augmentation on the 88 candidate areas from images C and D is presented in Appendix A, Table A2. As it can be seen from this Table, although Xception, VGG16, VGG19, ResNet50, Inception V3, InceptionResNetV2, DenseNet121, DenseNet169 and DenseNet201 are powerful and computationally intensive models, our less computationally intensive architecture provides much better results in the damaged tree categories problem.

On the other hand, the proposed model with data-augmentation achieves a high F1_score on class 1, 2, 3 and 4 with 92.75%, 89.86%, 89.66%, and 88.89, respectively. This can be explained by the fact that the model correctly distinguishes the color, shape, and texture of each one of the four classes.

The results of the detection on the input test areas C and D are shown in Figure 2.7 and 2.8. The red boxes present the selected areas produced by our candidate selection technique (see Section 4.4). Each red box has two numbers. The first number is the tree category estimated by the CNN models; C1, C2, C3 and C4 refer to the tree category 1, 2, 3 and 4 respectively. The second number indicates the probability calculated by our classification CNN model. Actually, this probability indicates the level of confidence of the model.

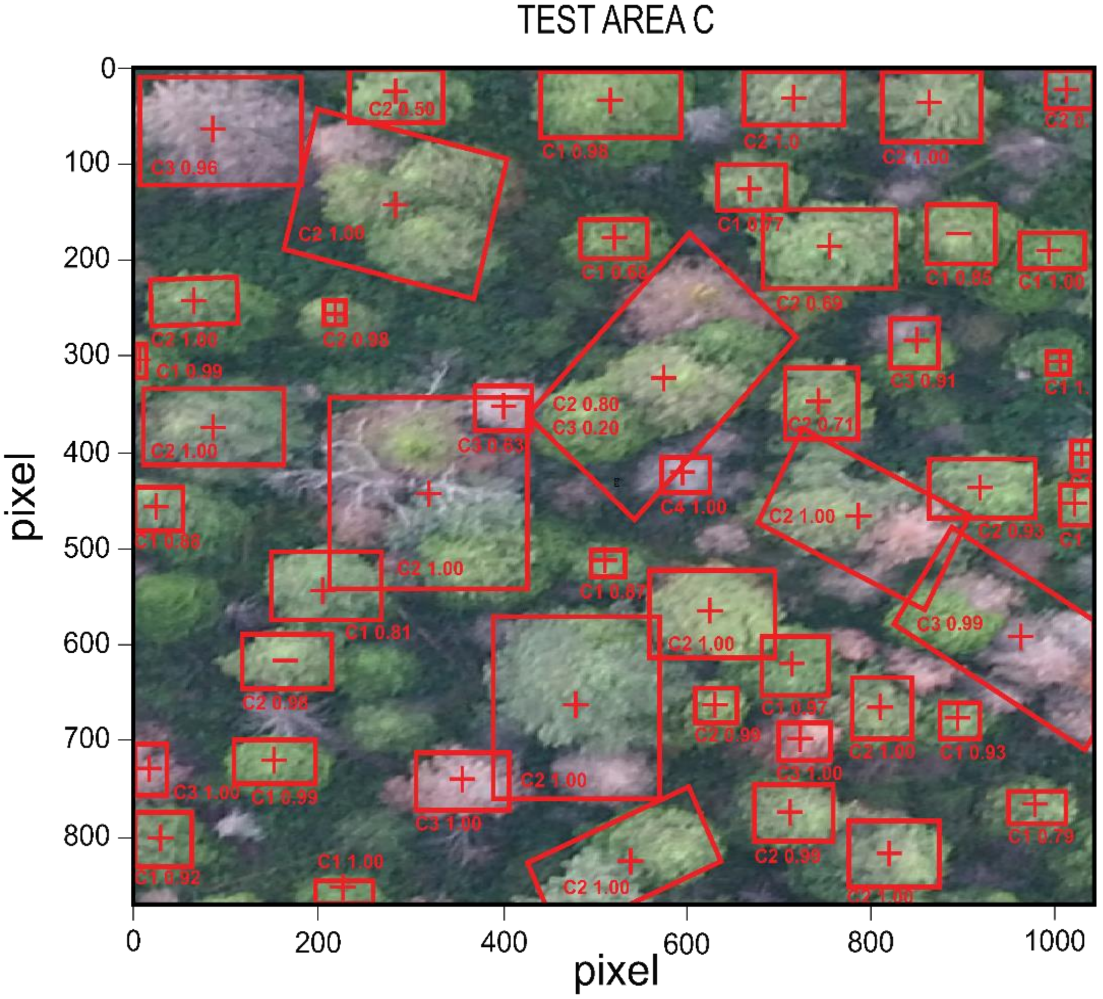


Figure 2.7. Results of the detection of damaged tree categories on test area C. C1, 2, 3 and 4 indicate the estimated class by our CNN classification model together with the corresponding probability. The symbols “+” and “-” indicate respectively correct and incorrect class estimation by our model.

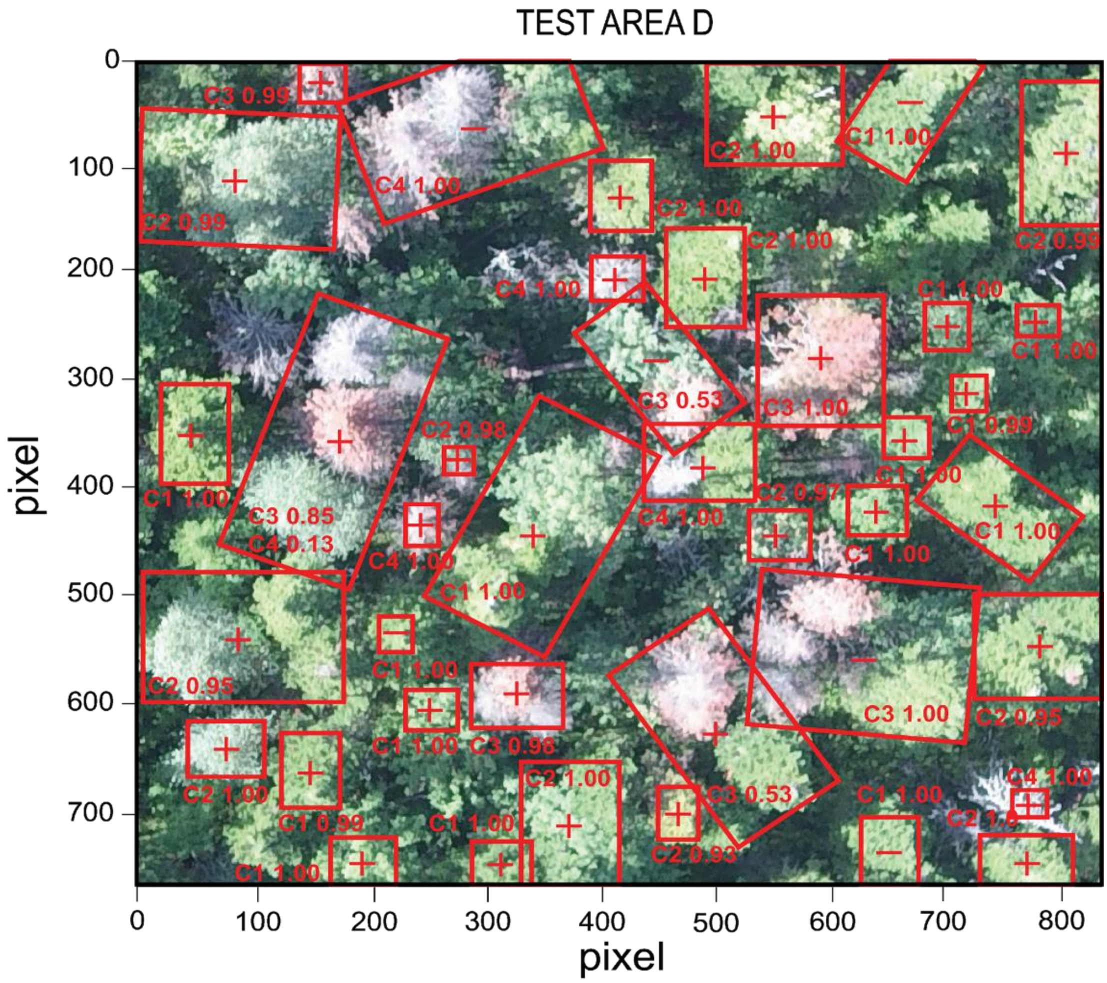


Figure 2.8. Results of the detection of damaged tree categories on test area D. C1, 2, 3 and 4 indicate the estimated class by our CNN classification model together with the corresponding probability. The symbols "+" and "-" indicate respectively correct and incorrect class estimation by our model.

As it can be seen from Figure 2.7 and 2.8 our model recognizes correctly the tree category in most candidate regions (red boxes). In the few cases where the boxes include more than one tree crown, the model correctly predicts the class that occupies the largest number of pixels in the area of the box.

6. Conclusion

Damage to forests with fir trees caused by the attacks of beetles is one of the main problems in forestry, especially in Central Siberia of Russia. An efficient method for recognizing categories of tree damage on ERS imagery will aid substantially eliminating fir trees colonized by *P. proximus* beetles, which will significantly reduce the consequences of its spread to new territories. The presented results are of great interest, both for scientific and practical purposes, since such work in the task of classifying categories of tree damage caused by beetle attacks on UAV images using CNN methods has not yet been encountered. Recognizing the categories of fir trees is a difficult task due to the fact that forests of Siberia are of high density.

In this chapter, we developed a model based on CNN to classify damage to fir trees caused by beetle attacks on RGB images acquired by UAV. Our network consists of six convolutional blocks. The network can recognize four categories of Siberian fir, from healthy to dry, with high prediction accuracy on images from different survey data. We have shown that our model provides better performance than the most powerful CNN models, such as Xception, VGG, ResNet, Inception, InceptionResNet, DenseNet. We showed that using data augmentation increased substantially the performance of our CNN model. We note that our model, trained with data augmentation, showed up to 98.77% accuracy for categories one, three and four in the test. The second category has the lowest recognition accuracy. This is due to the fact that we had to merge three categories (weakened trees, heavily weakened trees, and dying trees) into one category – trees colonized by the beetle, since it is presently not possible to observe the differences between these three categories on the imagery acquired in this study. Our model recognizes with higher accuracy the first, third, and fourth categories.

Regarding alternative models, the VGG16 model, on average, showed higher results, among other models considered in the paper. The VGG16 model recognized the first, second and fourth categories with an accuracy of 85.9%, 79.76% and 94.37%, respectively. Recognition of the third category with an accuracy higher than 88.89% is achieved by the ResNet50 model. The lowest test results in comparison with other considered models demonstrated the model InceptionResNetV2.

III. Olive Tree Biovolume from UAV Multi-Resolution Image Segmentation with Mask R-CNN

1. Introduction

Most of the world’s olive oil—around 2 million tones (66% of global production)—is produced in the European Union. Although recent intensification techniques organize olive trees in hedgerows [MaGM20], most olive groves are rainfed and trees are planted at ~6 m spacing. The main producers are Spain (66% of EU production), Italy (15%), Greece (13%) and Portugal (5%). Spain has a leading position in the world in the production of olive oil (43% of the global production). One of the needed tasks for the agricultural business is the automation of the assessment of the size and health condition of olive trees (*Olea europaea* L.) for further forecast of the yield and profit. In addition, there are emerging threats that should be urgently addressed: the spread of the infection with the bacterium *Xylella fastidiosa Wells* et al. (1987) [Scag19], and the effects of climate change such as the increase in extreme events (e.g., droughts, floods, and cold waves). These impacts affect photosynthesis, evapotranspiration, plant nutrition, and eventually plant growth and production [BDMC19, SMFD08]. Performing automatic monitoring of olive tree growth would be essential in these regions to effectively address these threats. Nowadays, the application of machine learning methods on very high spatial resolution satellite and aerial images opens the possibility of detecting isolated shrubs and trees at regional scale [BTKR20, GACP20, GBRT21, GTAC17, StKa20].

In precision agriculture, the use of unmanned aerial vehicle (UAV) images with the near infrared (NIR), red, green, and blue spectral bands has been successfully incorporated for monitoring plant growth and status [CRDD15, LNNS16]. Spectral indices such as the normalized difference vegetation index (NDVI) or green normalized difference vegetation index (GNDVI) can be used to determine the type of crop, its performance, and its ripening stage [CVVG18]. GNDVI index is more sensitive to variation in crop chlorophyll content than the NDVI index, and GNDVI also has a higher saturation threshold, so it can be used in crops with dense canopies or in more advanced development stages and to evaluate moisture content and nitrogen concentrations in plant leaves [GiKM96]. On the other hand, NDVI index is particularly suitable for estimating crop vigor during the initial development stages [BSVV19, CVVG18].

On the other hand, DL methods in general and CNNs in particular have demonstrated impressive results over classical methods in extracting spatial patterns from natural RGB-images. In fact, CNNs constitute the state-of-the-art in all the fundamental computer vision tasks, in image classification [KrSH12], object detection and instance segmentation [DSMB20, GIEM19, HoMe18, ZhTZ19]. A good approach to accurately estimate olive tree crowns is by using instance segmentation models such as Mask R-CNN [HGDG18], one of the most accurate CNN-based segmentation methods.

The main limitation of DL CNNs is that they require a large training dataset to achieve good results. In practice, in real world applications, several optimizations are used to overcome this limitation, namely, transfer learning, fine tuning, data augmentation [TPHH17], and potentially data-fusion.

The objective of this chapter is to illustrate the potential of deep CNNs for estimating the biovolume of olive-tree plantations from the tree crowns and shadows identified in ultra-high resolution images (less than a 30 cm, [ZQHF15]). We first trained CNNs to identify olive tree crown and shadow segments. Then, we approximated tree biovolumes from the tree crown surfaces and the tree heights inferred from the shadow lengths. Previous works on shrubs and trees mainly focused on detection of plant species or damaged stages in unmanned aerial vehicle (UAV) images [GBRT21, STAR19]. As far as we know, this is the first work in exploring the instance segmentation task for plant species segmentation with the objective of estimating the biovolume of trees.

The main contributions of this chapter can be listed as follows:

- We have built a new annotated multi-spectral orthoimages dataset for olive tree crown segmentation, called OTCS-dataset. OTCS-dataset is organized into four subsets of different spectral bands and vegetation indices (RGB, NDVI, and GNDVI), at two spatial resolutions (3 cm/pixel and 13 cm/pixel).

- We evaluated the instance segmentation Mask R-CNN model for the tasks of olive trees crown segmentation and shadows segmentation in UAV images. We present a model based on the fusion of RGB images and vegetation indices that improves segmentation over models without image fusion.

- We estimated the biovolume of olive trees based on the area of their crowns and their height inferred from their shadow length.

- Our results show that NDVI or GNDVI spectral indices information with 13 cm/pixel resolution are enough for accurately estimating the biovolume of olive trees.

This chapter is organized as follows. The related works are presented in Section 2. The materials and methods are shown in Section 3, where the study area and UAV RGB and multispectral images are in Section 3.1, the UAV RGB and multispectral images are in Section 3.2, the OTCSS-dataset construction is in Section 3.3, the Mask R-CNN is in Section 3.4, the experimental setup is in Section 3.5, the metrics for CNN performance evaluation are in Section 3.6, and the biovolume calculation from tree shadow estimations are in Section 3.7. The experimental results are presented in Section 4, where the tree crown and tree shadow segmentation with RGB and vegetation indexes are in Section 4.1 and the results of tree biovolume calculations are in Section 4.2. The conclusion is given in Section 5.

2. Related Works

Most problems in plant monitoring using high resolution RS data are formulated as either: (a) an image classification problem, (b) an object detection problem, (c) a semantic segmentation problem or (d) an instance segmentation problem. In image classification, the method analyzes a given input image and outputs a label that describes the object-class existent in that image (see illustration in Figure 3.1a). In object detection, the method analyzes the input image and outputs a label together with a bounding box that delimits where the object-class is located in the image (Figure 3.1b). In semantic segmentation, the method analyzes the input image and outputs a label together with a polygon that delimits the pixels of each object-class (Figure 3.1c). In instance segmentation, the method analyzes the input image and outputs a label together with a polygon that delimits the pixels of each instance of the object-class (Figure 3.1d). Therefore, instance segmentation methods are potentially more suitable for estimating the surface of olive-tree crowns as they provide a precise estimation of all the pixels that constitute each olive-tree individual.

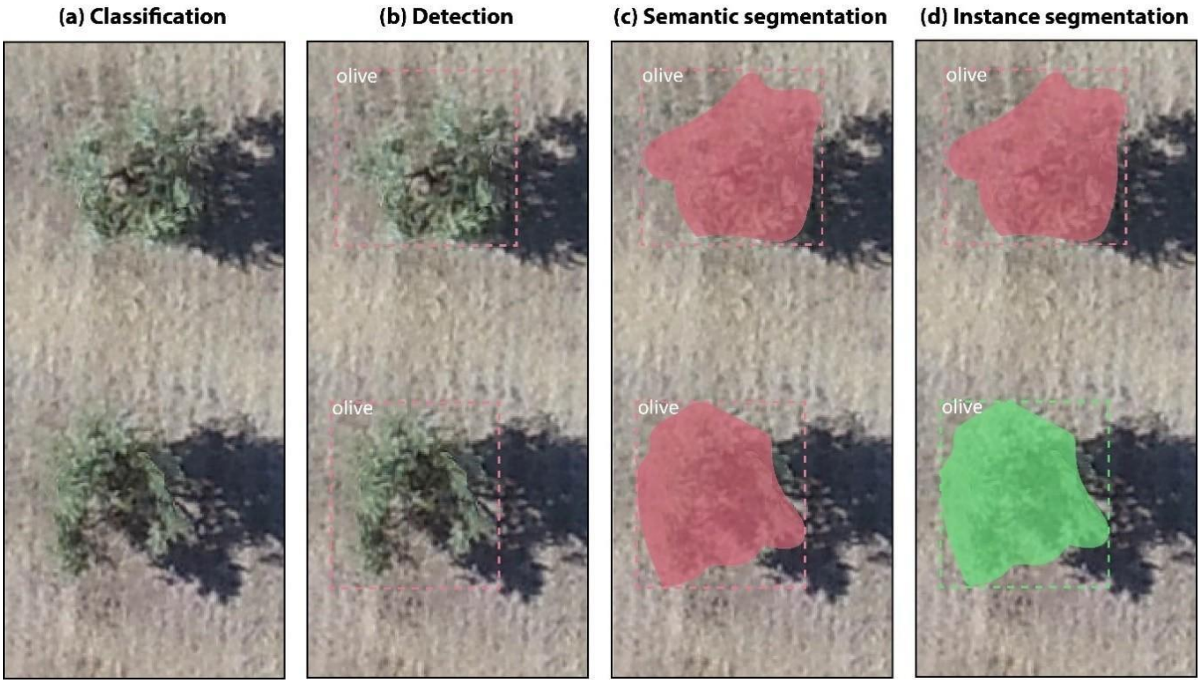


Figure 3.1. Illustration of the four fundamental computer vision tasks in the problem of olive-tree monitoring: (**a**) Image classification, (**b**) Object detection, (**c**) Semantic segmentation and (**d**) Instance segmentation.

Unfortunately, the majority of the existing plant monitoring works reformulate their problems as either image classification tasks [NaAV19, STAR19] or object detection tasks [CBMO19, CCJL18, FLGX18, KMGO19, NeHH19, ReFa18, SMAD19, WWSD19]. For example, the authors in [OnIs18] showed that applying a simple CNN-pixel-wise classification model on the fusion of high resolution digital surface model (DSM) with NDVI radiometric index provides a good potential for estimating crop/soil surface.

Few works address precision agriculture problems using DL segmentation methods. For example, for the estimation of pomegranate tree canopy in UAV images, the authors in [ZYNW18] compared the performance of two CNN-based segmentation models, U-Net and Mask RCNN. Their experiments showed that faster RCNN achieved better results with respect to U-Net, with a mean average precision (mAP) of 57.5% versus 36.2%. In [GACP20] the authors showed that the fusion of Mask-Fast RCNN and OBIA methods increases by 25% the overall accuracy of the segmentation of scattered shrubs in UAV, airborne and GoogleEarth imagery. In [LQNE20] the authors evaluated the performance of five CNN-based methods for the semantic segmentation of a single endangered tree species, called *Dipteryx alata* Vogel, in UAV images. In particular, they evaluated SegNet, U-Net, FC-DenseNet, and two DeepLabv3+ variants and found that FC-DensNet overcomes all the previous methods with an overall accuracy of 96.7%. In [GuKN19], the authors developed a CNN based semantic segmentation method inspired by U-Net for the detection of mango tree individual crowns. Their experiment showed an overall accuracy of the order of 90%.

In the present chapter, we will estimate olive tree biovolume from the tree crowns and tree shadows obtained by applying Mask R-CNN instance segmentation on ultra high resolution UAV images. Currently, Mask R-CNN is considered one of the most accurate deep CNN-based methods.

3. Materials and Methods

3.1. Study Area and UAV RGB and Multispectral Images

The study area is located in Andalusia, Spain (37°23′57″ N 3°24′47″ W). The climate is Mediterranean, characterized by severe summer droughts and mild-wet winters. Average total annual precipitation is 400 mm and mean annual temperature is 15 ºC. This area is dominated by rainfed cereal croplands and olive groves in flatlands with some patches of natural vegetation in hills (Figure 3.2). To avoid competition for water availability among olive trees, they are separated by about 6 m from each other. The test area is within an olive grove of 50 hectares comprising 11,000 trees that were planted in 2006. We used a flat rectangle of 560 m × 280 m containing approximately 4000 trees as our study object.

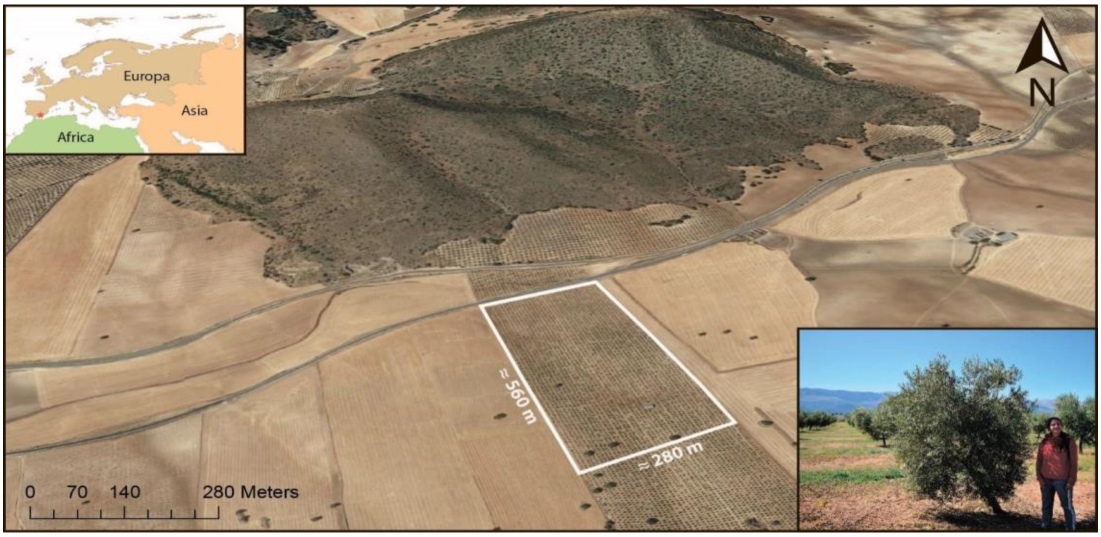


Figure 3.2. The test area in Andalusia, southern Spain (37°23′57″ N 3°24′47″ W).

3.2. UAV RGB and Multispectral Images

To compare the effect of DL models on different spatial and spectral resolutions, we made two UAV flights at 120 m height that captured an RGB image at ultra-high spatial resolution, and a multispectral image at very-high resolution:

(1) In February 2019, we flew a Sequoia multispectral sensor installed on the Parrot Disco-Pro AG UAV (Parrot SA, Paris, France) that captured four spectral bands (green, red, red edge, and near-infrared -NIR). The spatial resolution of the multispectral image was 13 cm/pixel. We then derived the vegetation indices detailed in the introduction: the normalized difference vegetation index (NDVI) (3.1) [Meas19], and the green normalized difference vegetation index (GNDVI) (3.2) [GiKM96].

$$NDVI = \frac{NIR-Red}{NIR+Red}, \tag{3.1}$$

$$GNDVI = \frac{NIR-Green}{NIR+Green}. \tag{3.2}$$

(2) In July 2019, to get finer spatial resolution, we flew the native RGB Hasselblad 20-megapixel camera of the DJI-Phantom 4 UAV (Parrot SA, Paris, France). The spatial resolution of the RGB image was 3 cm/pixel. These RGB images were then converted to 13-cm/pixel by spatial

averaging so they could be compared to. In both flights, images were donated by the company Garnata Drone S.L. (Granada, Spain).

The specific conditions for the present data acquisition are weather conditions (sunny and cloudless day) and the time of shooting before sunset. For example, in our study, the following shots were made at 10:51, 9 February 2019 and at 18:54, 19 June 2019 (sunset on that day is at 20:27).

3.3. OTCSS-Dataset Construction

To build a dataset for the task of instance segmentation of olive tree crowns and tree shadows that could let us assess the effect of decreasing spatial resolution and of gaining spectral information, we produced four subsets of data: (a) RGB-3, (b) RGB-13, (c) NDVI-13 and (d) GNDVI-13, where 3 and 13 indicate the spatial resolution of the images in cm/pixel (Figure 3.3).

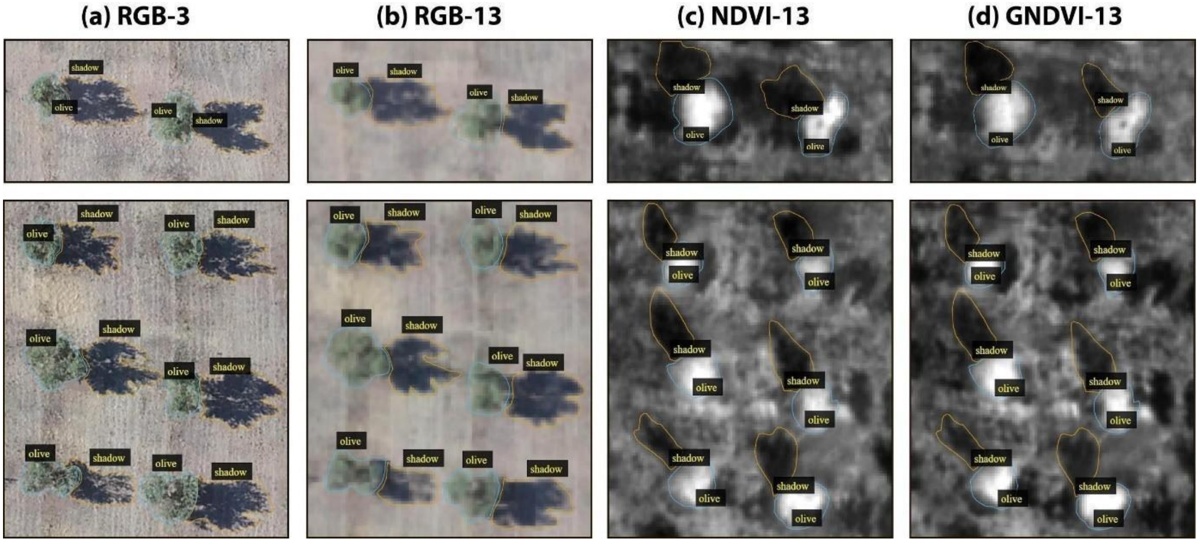


Figure 3.3. Examples of two image patches (first and second rows) in the four subsets of images (four columns) used to assess the effect of decreasing spatial resolution (RGB-3 versus RGB-13) and gaining spectral information (RGB-13 versus NDVI-13 OR GNDVI-13) for the task of instance segmentation of olive tree crowns and shadows in the OTCSS-dataset. (a) RGB-3 cm/pixel, (b) RGB-13 cm/pixel, (c) NDVI-13 cm/pixel and (d) GNDVI-13 cm/pixel. RGB: Red, Green, Blue; NDVI: normalized difference vegetation index; GNDVI: green normalized difference vegetation index.

For each subset of data, we prepared 150 image patches that contained 2400 trees, of which 120 images (80% of the dataset) were used for training the model, and 30 images (20%) were used for testing the model on the olive tree crown class (Table 3.1) and on the olive tree shadow class (Table 3.2).

Table 3.1. A brief description of the number of image patches and segments in the four subsets of the Olive Tree Crown Segmentation in the OTCSS-dataset: RGB-3 cm/pixel, RGB-13 cm/pixel, NDVI-13 cm/pixel, and GNDVI-13 cm/pixel. RGB: Red, Green, Blue; NDVI: normalized difference vegetation index; GNDVI: green normalized difference vegetation index.

Tree Crown Subset	# of Training Images	# of Training Segments	# of Testing Images	# of Testing Segments	Total of Images	Total of Segments
RGB-3	120	480	30	120	150	600
RGB-13	120	480	30	120	150	600
NDVI-13	120	480	30	120	150	600
GNDVI-13	120	480	30	120	150	600
Total	**480**	**1920**	**120**	**480**	**600**	**2400**

Table 3.2. A brief description of the number of image patches and segments in the four subsets of the Olive Tree Shadow Segmentation in the OTCSS-dataset: RGB-3 cm/pixel, RGB-13 cm/pixel, NDVI-13 cm/pixel, and GNDVI-13 cm/pixel. RGB: Red, Green, Blue; NDVI: normalized difference vegetation index; GNDVI: green normalized difference vegetation index.

Tree Shadow Subset	# of Training Images	# of Training Segments	# of Testing Images	# of Testing Segments	Total of Images	Total of Segments
RGB-3	120	480	30	120	150	600
RGB-13	120	480	30	120	150	600
NDVI-13	120	480	30	120	150	600
GNDVI-13	120	480	30	120	150	600
Total	**480**	**1920**	**120**	**480**	**600**	**2400**

Each image patch contained from one to eight olive trees with their corresponding tree crowns and tree shadows (see the example in Figure 3.3).

The general scheme of creating the data set is shown in Figure 3.4. The original UAV images were mosaicked into an orthophoto by using Pix4D 4.0. QGIS 2.14.21 was used for reducing the spatial resolution of the RGB-3 cm/pixel to the resolution of RGB-13 cm/pixel, and for calculating the NDVI and GNDI indices. ENVI Classic was used for creating the patches and converting them from .tiff to .jpg format (the most suitable format for training DL models). During the .tiff to .jpg conversion the spatial resolution was artificially increased to 13 cm/pixel by QGIS 2.14.21 program. For creating and annotating the tree crown and the tree shadow segments in each image patch, we used VGG Image Annotator 1.0.6. which is a standalone software for manual annotation of images. The annotation process for this instance segmentation task was completely manual. That is, the annotator created a polygon surrounding each olive tree crown and another polygon surrounding each tree shadow instance. The created class labels with VGG annotator were then saved in a JSON format.

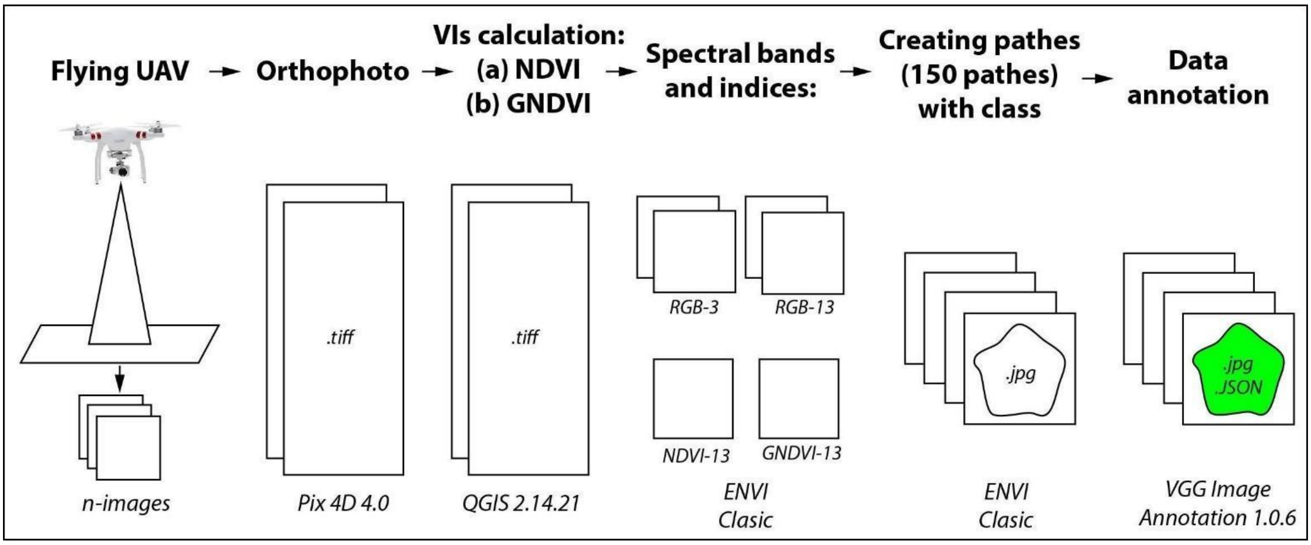


Figure 3.4. The process of preparing the images of OTCS-dataset.

3.4. Mask R-CNN

The task of locating and delimiting all the pixels that constitute each individual olive tree crown in UAV images is called instance segmentation. This task is one of the most complex problems in computer vision. In this work we used the modern Mask R-CNN network (regions with convolutional neural networks) [GBRT21], which extends the faster R-CNN detection model [HZRS15]. Mask R-CNN analyzes an input image and provides three outputs for each object-class: (1) a class label that indicates the name of the object-class, (2) a bounding box that delimits each object-class and (3) a mask that delimits the pixels that constitute each object-class. For the considered problem in this work, Mask R-CNN generates for each olive tree a binary mask (with values 0 and 1), where value 1 indicates an olive tree pixel and 0 indicates a non-olive tree pixel.

Mask R-CNN is based on a classification model for the task of feature extraction. In this work, we used ResNet50 CNN [GBRT21] to extract increasingly higher-level features from the lowest to the deepest layer levels.

To further improve the generalization capacity of the segmentation model, we assessed the effect of the data augmentation technique [Coco21], which consists of increasing the size of the dataset by applying simple transformations such as cropping (i.e., removing columns/rows of pixels at the sides of images), scaling, rotation, translation, horizontal and vertical shear. Instead of training Mask R-CNN (based on ResNet50) from scratch on our dataset, we used transfer learning, which consists of first initializing the weights of the model with pre-trained weights on a well-known COCO-dataset, then retraining the model on our own dataset. The process of retraining the last years on a small new dataset is called fine tuning [TPHH17].

3.5. Experimental Setup

The preprocessing and training stages were carried out using Python programming language, version 3.5.2, and TensorFlow Object Detection API [Tens20], an open-source software library for high-performance DL models. The calculations were performed on a computer with an Intel Xeon E5-2630v4 processor, accelerated using an NVIDIA Titan Xp graphics processor as a platform for learning and testing the proposed methodology. We used a learning rate of 0.001 and the stochastic gradient descent solver as an optimization algorithm. We trained Mask R-CNN network for 100 to 150 epochs on each different spectral bands and indices, i.e., RGB-3, RGB-13, NDVI-13, and GNDVI-13

Thanks to transfer-learning from COCO and fine-tuning, the execution time of the training process of Mask R-CNN on our dataset takes about half an hour on the GPU and several hours on the CPU. Testing Mask R-CNN over test images is very fast, almost real-time.

Several experiments were carried out to assess the effect of pixel size and the effect of using vegetation indices (that incorporate NIR information) instead of RGB images. We also quantified the benefit of using data augmentation on a small dataset. In total, we trained the following Mask R-CNN models:

- For tree crown estimation, we trained models on each subset of data separately (i.e., RGB-3, RGB-13, NDVI-13, and GNDVI-13) without (group A of models) and with data augmentation (group B of models) (i.e., scaling, rotation, translation, horizontal and vertical shear). In addition, we also tested whether data fusion could improve the generalization of the final model, that is, whether training a single model (model C) on all the RGB, NDVI, and GNDVI data together at 13 cm/pixel could result in a single general model able of accurately segmenting olive tree crowns independently of the input (i.e., RGB-13, NDVI-13, or GNDVI-13).

- For tree shadow estimation, we just trained one model (model D) with data augmentation on the RGB-3 subset to estimate tree heights on the dataset with highest spatial resolution precision. That model was then applied to the four subsets of data. In addition, we also tested whether data fusion could improve the generalization of the final model, that is, whether training a single model (model E) on all the RGB, NDVI, and GNDVI data together at 13 cm/pixel could result in a single general model able of accurately segmenting olive tree shadows independently of the input (i.e., RGB-13, NDVI-13, or GNDVI-13).

3.6. Metrics for CNN Performance Evaluation

To evaluate the performance of the trained Mask R-CNN on OCTS-dataset in the task of olive tree crown and shadow instance segmentation, we used the F1-score metric, which is defined as the harmonic mean of the precision and recall [Sasa07].

Mask R-CNN produces three outputs, a bounding-box, a mask, and a confidence about the predicted class. To determine whether a prediction is correct, the Intersection over union (IoU) or Jaccard coefficient [RTGS19] was used. It is defined as the intersection between the predicted bounding-box and actual bounding-box divided by their union. A prediction is true positive (TP) if IoU > 50%, and false positive (FP) if IoU < 50%. IoU is calculated as follows (3.3):

$$IoU = \frac{Area\ of\ Overlap}{Area\ of\ Union}. \tag{3.3}$$

Usually, a threshold value of 0.5 is used, as it usually shows high indicators of scores [HGDG18]. The precision (3.4) and recall (3.5) are calculated as follows:

$$Precision = \frac{TP}{TP+FP} = \frac{TP}{\#ground\_truths}, \tag{3.4}$$

$$Recall = \frac{TP}{TP+FN} = \frac{TP}{\#predictions}. \tag{3.5}$$

Precision determines the percentage of correctly recognized labels and Recall is part of a successful extraction of relevant labels.

F1-score is the weighted average of precision and recall (3.6). It takes both false positives and false negatives into account to ultimately measure the global accuracy of the model:

$$F1 = \frac{2\times Precision\times Recall}{Precision\times Recall}. \tag{3.6}$$

3.7. Biovolume Calculation from Tree Crown and Tree Shadow Estimations

To estimate tree biovolumes from the tree crown and tree shadow polygons retrieved from the Mask R-CNN models outputs, we approximated it to a cylinder with a base of equal perimeter to the polygon of the tree crown and with a height equal to the height of the tree estimated from the length of its shadow minus 0.5 m corresponding to the height of the unbranched trunk:

- For the tree crown surface (S), we first obtained the perimeter (P) of the tree crown polygon and then calculated the surface of a circle of the same perimeter.
- For tree height (h), we followed [BaHa91] to derive tree heights from tree shadows. In a flatland, the height of the tree (h) can be calculated from the length of the shadow (L) and the

angle (θ) between the horizon and the sun's altitude in the sky. The tree shadow length was derived from the shadow polygons as the distance from the tree crown polygon to the far end of the shadow polygon using QGIS 2.14.21 program. The angle between the horizon and the sun's altitude can be calculated from the geographical position (latitude and longitude), date and time of imagery acquisition (3.7) [WBPS12]. Since the fly time and date of DL-Phantom 4 Pro drone was 10:51, 9 February 2019, and the coordinates were 37°23′57″ N 3°24′47″ W, the θ was 29.61°. The fly time and date of Parrot Disco-Pro AG was 18:54, 19 June 2019, and the coordinates were 37°23′57″ N 3°24′47″ W, the θ was 26.22° [Sunc20]:

$$h = L \times tan\ (\theta). \tag{3.7}$$

- Finally, for tree canopy volume (V), we approximated the biovolume in m3 by multiplying the tree crown surface (S) in m2 by the tree height minus 0.5 m (L-0.5) in m. We systematically removed 0.5 m to the tree height to exclude the lower part of the tree trunk, on which there are no branches (on average about 0.5 m in height) (Figure 3.5). Though we could only take six ground truth samples for canopy biovolume, we assessed the overall accuracy of it as follows:

$$Accuracy = \left(1 - \sum_{i=1}^{N} \frac{|V_{Gi} - V_{Mi}|}{V_{Gi}}\right) \times 100\%. \tag{3.8}$$

where VG is the approximate volume of tree canopy estimated from ground truth measurements, VM is the approximate volume of the tree canopy derived from the Mask R-CNN segmentation of tree crowns and shadows, i is each individual tree, and N is the total number of trees.

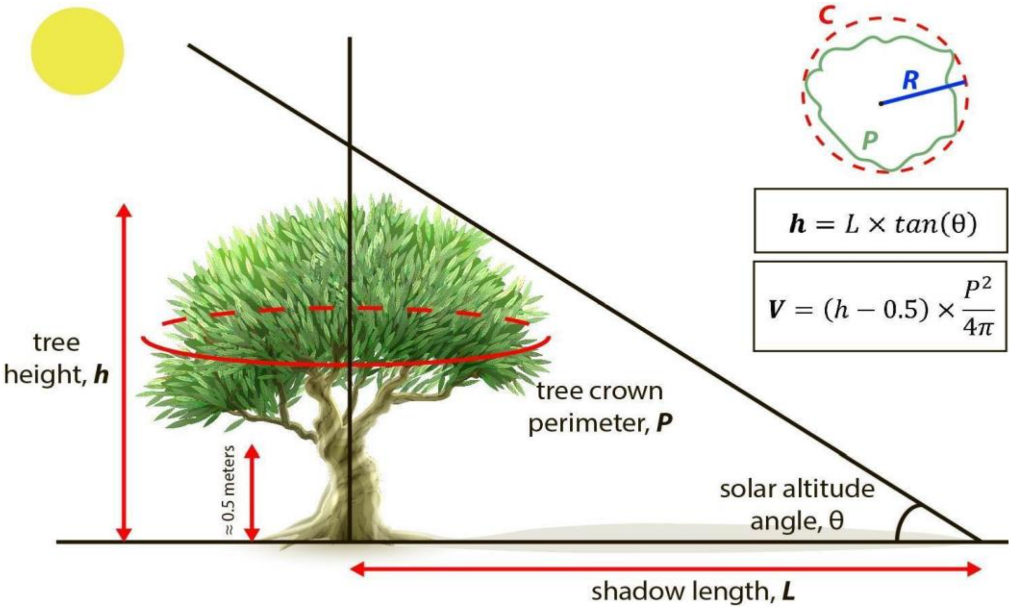


Figure 3.5. Illustration of the approximated biovolume in olive trees estimated from the automatic retrieval of tree crown and tree shadow polygons from DNN Mask R-CNN applied on RGB or multispectral Unmanned Aerial Vehicle imagery. The volume of the tree canopy was approximated to a cylinder with a base of equal perimeter (P = C) to the polygon of the tree crown and with a height equal to the height (h) of the tree estimated from the length (L) of its shadow minus 0.5 m corresponding to the height of the unbranched trunk at the bottom. V: approximated biovolume; P: tree crown perimeter, equal to the circumference (C) of the cylinder base; L: length of tree shadow; θ: solar altitude angle; h: tree height. The olive tree picture was designed by macro vector and downloaded from www.freepik.com.

4. Experimental Results

This section has been divided into two parts. The segmentation results of the RGB and vegetation indices images are shown in Section 4.1. The results of tree biovolume calculations are presented in Section 4.2.

4.1. Tree Crown and Tree Shadow Segmentation with RGB and Vegetation Indices Images

The performance, in terms of precision, recall, and F1-score, of all Mask R-CNN models on the corresponding test subsets of data are shown in Table 3.3 for tree crowns and in Table 3.4 for tree shadows.

Table 3.3. Segmentation performance of Mask R-CNN models for "Olive tree crown" class applied to the four subsets of the OTCSS-dataset in terms of Precision, Recall and F1-measure. TP: True Positive; FP: False Positive; FN: False Negative. The testing datasets were: RGB-3 cm/pixel, RGB-13 cm/pixel, NDVI-13 cm/pixel, and GNDVI-13 cm/pixel. RGB: Red, Green, Blue; NDVI: normalized difference vegetation index; GNDVI: green normalized difference vegetation index.

Testing Subset	TP	FP	FN	Precision	Recall	F1
A. Trained models on each subset without data augmentation						
RGB-3	120	0	0	1.0000	1.0000	**1.0000**
RGB-13	119	0	1	1.0000	0.9916	**0.9958**
NDVI-13	114	2	6	0.9827	0.9500	0.9660
GNDVI-13	110	0	10	1.0000	0.9166	**0.9564**
B. Trained models on each subset with data augmentation						
RGB-3	120	0	0	1.0000	1.0000	**1.0000**
RGB-13	118	0	2	1.0000	0.9833	0.9915
NDVI-13	118	13	2	0.9007	0.9833	0.9401
GNDVI-13	118	12	2	0.9076	0.9833	0.9439
C. Trained models on the fusion of all 13-cm/pixel subsets of images and with data augmentation						
RGB-13	119	0	1	1.0000	0.9916	**0.9958**
NDVI-13	116	0	4	1.0000	0.9666	**0.9830**
GNDVI-13	109	0	11	1.0000	0.9083	0.9519

Graphical examples of the segmentation results of olive tree crowns and tree shadows are presented in Figure 3.6.

As shown in Table 3.3 for tree crown segmentation, all trained and tested Mask R-CNN models showed high F1 score, above 94% across all subsets of data. Data augmentation did not significantly affect the F1 score. The best performance (F1 = 100%) was reached with the RGB subset at a spatial resolution of 3 cm/pixel.

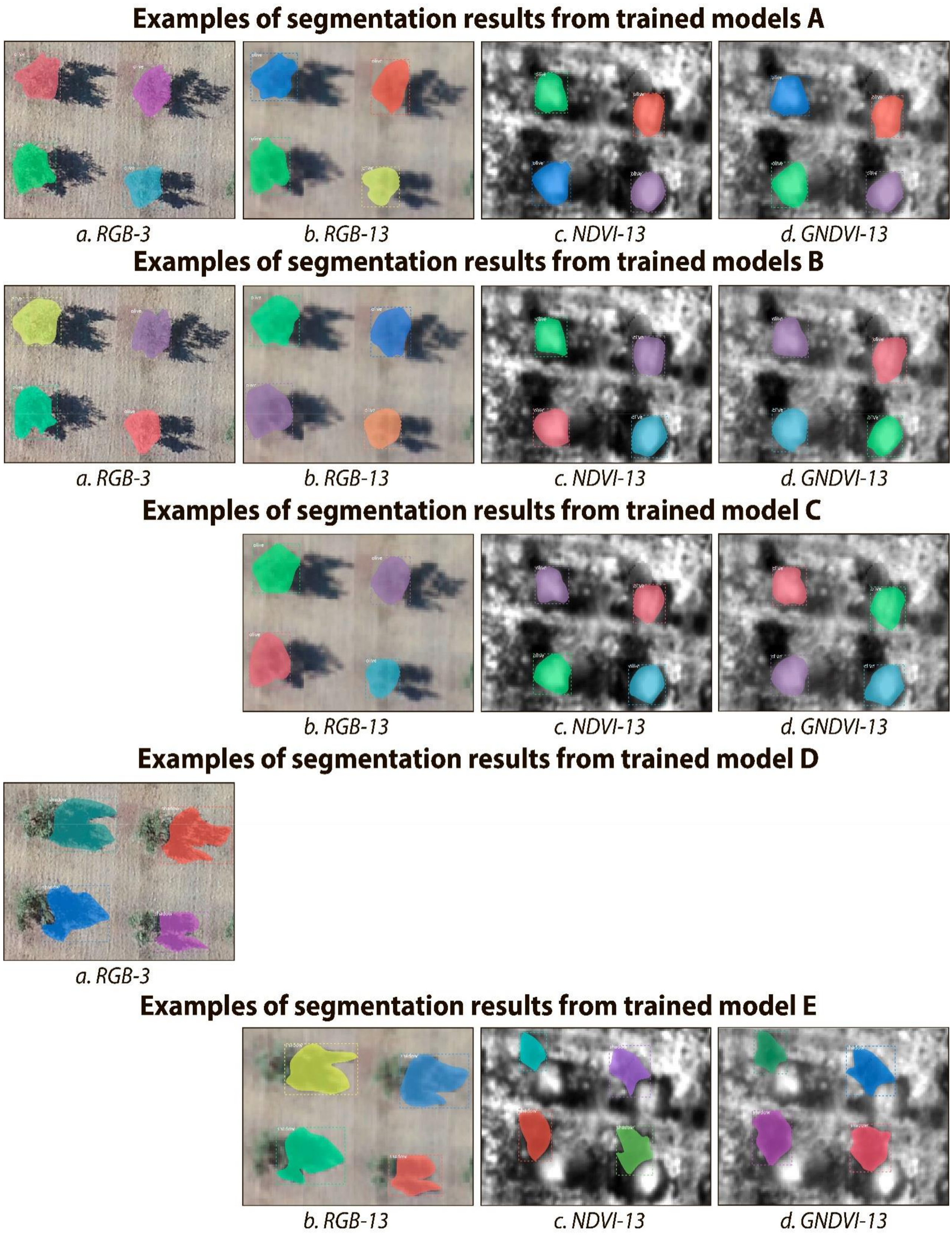

Examples of segmentation results from trained models A
a. RGB-3
b. RGB-13
c. NDVI-13
d. GNDVI-13
Examples of segmentation results from trained models B
a. RGB-3
b. RGB-13
c. NDVI-13
d. GNDVI-13
Examples of segmentation results from trained model C
b. RGB-13
c. NDVI-13
d. GNDVI-13
Examples of segmentation results from trained model D
a. RGB-3
Examples of segmentation results from trained model E
b. RGB-13
c. NDVI-13
d. GNDVI-13

Figure 3.6. Examples of the segmentation results for the class “Olive tree crowns” (models A, B and C) and for the class “Olive tree shadows” (models D and E) using Mask R-CNN in the four image subsets of the OTCSS-dataset. See Section 3.4. Experimental Setup for model explanation. The testing datasets were: (a) RGB-3 cm/pixel, (b) RGB-13 cm/pixel, (c) NDVI-13 cm/pixel and (d) GNDVI-13 cm/pixel. RGB: Red, Green, Blue; NDVI: normalized difference vegetation index; GNDVI: green normalized difference vegetation index.

For the RGB datasets, coarsening the pixel size from 3 to 13 cm/pixel slightly decreased F1 by 0.42% without data augmentation (models A) and by 0.86% with data augmentation (models B). At

13-cm/pixel resolution, the 3-band RGB images always produced greater F1 scores than the single-band NDVI or GNDVI images. However, the model trained with data fusion (model C, which is trained on RGB, NDVI, and GNDVI images altogether) showed equivalent or greater F1 than the models trained without data fusion (both with and without data augmentation, models A and B). For the NDVI-13 dataset, data fusion increased F1 score by 1.76% while data augmentation decreased it by 2.68%, compared to training just with the NDVI-13 dataset and without data augmentation, respectively. The F1 score reached on the GNDVI dataset was equivalent or greater than on the NDVI dataset.

Table 3.4. Segmentation performance of Mask R-CNN models for the "Olive tree shadow" class applied to the four subsets of the OTCSS-dataset in terms of Precision, Recall and F1-measure. TP: True Positive; FP: False Positive; FN: False Negative. The testing datasets were: RGB-3 cm/pixel, RGB-13 cm/pixel, NDVI-13 cm/pixel, and GNDVI-13 cm/pixel. RGB: Red, Green, Blue; NDVI: normalized difference vegetation index; GNDVI: green normalized difference vegetation index.

Testing Subset	TP	FP	FN	Precision	Recall	F1
D. Trained models on each subset with data augmentation						
RGB-3	120	0	0	1.0000	1.0000	**1.0000**
E. Trained models on the fusion of all 13-cm/pixel subsets of images and with data augmentation						
RGB-13	119	0	1	1.0000	0.9916	**0.9958**
NDVI-13	111	0	9	1.0000	0.9250	**0.9610**
GNDVI-13	117	0	3	1.0000	0.9750	**0.9873**

As shown in Table 3.4 for tree shadow segmentation, all trained and tested Mask R-CNN models show a high F1 score—above 96%. The highest F1 score was reached for the model (model D) trained and tested on RGB images at 3 cm/pixel. However, the data fusion model (model E, which is trained on RGB, NDVI, and GNDVI images altogether) also showed very high F1 on RGB-13 cm/pixel images (99.58%). The data fusion model (model E) performed better when tested on the RGB-13 (99.58%) and GNDVI-13 (98.73%) than on the NDVI-13 (96.10%) dataset for tree shadow segmentation.

4.2. Results of Tree Biovolume Calculations

Table 3.5 presents an example for the six olive trees that could be measured in the field for the approximation of free canopy volume from the tree perimeter and tree height segmentation obtained with the Mask R-CNN trained models. The overall accuracy was 94.51%, 75,61%, 82.58%, and 77,38% for RGB-3, RGB-13, NDVI-13, and GNDVI-13, respectively. The model trained and tested on RGB images at 3cm/pixel showed the highest overall accuracy for biovolume estimation. At 13 cm/pixel scale, the data fusion model also performed well and reached better accuracy on the NDVI subsets than on the GNDVI or RGB subsets.

Table 3.5. The averaged characteristics by best trained models for 6 test olive trees, where P is the perimeter of the tree crown polygon used as the circumference of the cylinder base, h is the tree height derived from the tree shadow, L is the tree shadow length, V is the approximate volume of the tree canopy. P, L, and h are expressed in m; V is in m3. Models A (tree crown) and D (tree shadows) were trained and tested on RGB 3 cm/pixel images. Models C (tree crown) and E (tree shadow) were trained on a data fusion of the RGB, NDVI, and GNDVI altogether at 13 cm/pixel images but tested separately on each subset of data at 13 cm/pixel.

Ground Truth				Models A & D				Models C & E				Models C & E				Models C & E			
				Tested on RGB-3				Tested on RGB-13				Tested on NDVI-13				Tested on GNDVI-13			
No.	**P**	**h**	**V**	**P**	**L**	**h**	**V**	**P**	**L**	**h**	**V**	**P**	**L**	**h**	**V**	**P**	**L**	**h**	**V**
1	6.3	2.5	**6.31**	6.6	4.3	2.4	**6.70**	7.1	4.1	2.3	**7.34**	7.7	3.6	1.8	**6.00**	9.4	3.6	1.8	**8.95**
2	6.5	2.6	**7.06**	6.5	4.8	2.7	**7.40**	8.0	4.3	2.4	**9.89**	8.2	4.5	2.2	**9.18**	8.2	4.5	2.2	**9.18**
3	8.3	3.0	**13.70**	8.8	4.6	2.6	**13.02**	10.0	5.8	3.3	**22.25**	10.0	5.2	2.6	**16.4**	10.6	5.2	2.6	**18.42**

Continuation of Table 3.5. The averaged characteristics by best trained models for 6 test olive trees, where P is the perimeter of the tree crown polygon used as the circumference of the cylinder base, h is the tree height derived from the tree shadow, L is the tree shadow length, V is the approximate volume of the tree canopy. P, L, and h are expressed in m; V is in m3. Models A (tree crown) and D (tree shadows) were trained and tested on RGB 3 cm/pixel images. Models C (tree crown) and E (tree shadow) were trained on a data fusion of the RGB, NDVI, and GNDVI altogether at 13 cm/pixel images but tested separately on each subset of data at 13 cm/pixel.

Ground Truth				Models A & D				Models C & E				Models C & E				Models C & E			
				Tested on RGB-3				Tested on RGB-13				Tested on NDVI-13				Tested on GNDVI-13			
No.	**P**	**h**	**V**	**P**	**L**	**h**	**V**	**P**	**L**	**h**	**V**	**P**	**L**	**h**	**V**	**P**	**L**	**h**	**V**
4	8.5	3.0	**14.37**	8.5	5.2	2.9	**14.11**	8.7	5.1	2.9	**14.34**	9.1	4.8	2.4	**12.28**	10.6	4.8	2.4	**16.66**
5	8.1	2.9	**12.53**	8.1	5.4	3.1	**13.41**	8.1	5.9	3.4	**14.89**	8.4	4.5	2.2	**9.63**	9.2	4.5	2.2	**11.56**
6	8.7	3.0	**15.05**	8.4	5.9	3.3	**16.02**	8.5	5.1	2.9	**13.78**	9.2	5.0	2.5	**13.21**	10.1	5.0	2.5	**15.93**

5. Conclusion

The assessment of tree size with UAV imagery under the framework of precision agriculture could help the automatic monitoring of tree growing and performance, with large economic implications as in the case of olive production. Our results show how applying Mask R-CNN, both on RGB and vegetation indices imagery and both at 3 and 13 cm/pixel, can be used to accurately (F1 always greater than 96%) map the crown and shadow segments of olive trees. These two polygons can then be used to estimate tree crown surface and tree height, two parameters commonly used to approximate tree canopy biovolume. Our test on six olive trees suggests that tree canopy biovolume can also be approximated (accuracy ranging from 77 to 95%) from these two CNN-derived parameters.

Currently, there are many affordable RGB and multispectral cameras that can be mounted on multi-rotor or fixed-wing drones and whose imagery can be automatically processed with CNN models for this purpose. On the one hand, RGB cameras mounted on a multi-rotor drone can capture much finer spatial resolution imagery, which increases accuracy of CNN models [GBRT21], but covering smaller areas (due to battery limitations), which results in more expensive imagery per hectare. On the other hand, multispectral cameras mounted on fixed-wing drones can capture coarser spatial resolution imagery but on larger areas, which decreases the cost per hectare, and with the benefit of incorporating plant reflectance in the near-infrared, and red-edge, which better relate to photosynthetic activity than just RGB [YJCA19]. Fusing both sources of data could join the advantage of both approaches, i.e., increase CNN accuracy, decrease the cost per hectare, and incorporate photosynthetic activity information [CLGJ17]. Our results show that CNN models trained and tested at much finer resolution (i.e., RGB at 3 cm/pixel) reached slightly greater accuracy (only 0.42% more) than at coarser resolution (i.e., RGB at 13 cm/pixel). More importantly, our results show that training CNN models on the fusion of all RGB, NDVI and GNDVI subsets of images at coarser resolution (i.e., 13 cm/pixel resolution) enables to have a generalized model that maintains very high accuracies (always greater than 95% and 96% for tree crown and tree shadow, respectively) no matter the nature of the image (RGB, NDVI or GNDVI) used in the testing. This generalization opens the possibility of using fixedwing multispectral or RGB imagery over extensive areas at a lower cost per hectare for the purpose of tree volume monitoring, with wide implications in precision agriculture, precision forestry and precision restoration.

Most sensors to obtain multispectral UAV imagery, such as the Parrot Sequoia used in this work, have four bands, i.e., green, red, red-edge, and near-infrared, but do not include a blue band to produce an RGB image [HKGP17]. Our results show that despite the absence of an RGB image, CNNs can reach very high accuracies just using the vegetation indices information (e.g., NDVI and GNDVI), if they are previously trained using a data fusion approach that incorporates both RGB and vegetation indices images. In other words, with a model trained in this way (RGB + NDVI + GNDVI) we could obtain greater precision in indices such as GNDVI which are usually obtained in flights with UAVs for precision agriculture. Furthermore, vegetation indices are widely used in agriculture around the world [CVVG18, HHST18].

It is important to note that data augmentation when applying to the Mask R-CNN model did not affect the results much, and even tended to slightly decrease the F-1 score [VaBC20]. The best results among datasets with a resolution of 13 cm/pixel were achieved by models trained on the RGB image dataset, which may indicate that the model works best on three-band images, in contrast to single-band ones as with NDVI and GNDVI vegetation indices [UBSA20]. This can be explained by the fact that the augmentation data gave us some objects similar to the weeds that grow below and among the olive trees, which caused false positives and decreased the final F1. Despite this, our proof of concept shows how the method of pixel segmentation using deep CNNs can be used with high efficiency in problems of agriculture and forestry on UAV images.

Our illustration of how the CNN segmentation results of tree crown and tree shadow can be used to approximate biovolume in several trees is encouraging to investigate further in this sense to improve the method. The calculated values correspond well with the ground measurements of the test trees, showing minimum error of 5.4%. Additional field measurements, calculations, and experiments are needed to get a better understanding of the prospects of this approach, which is a task of further studies. In the future work, it is planned to conduct testing of trained CNN on satellite data of medium resolution, which is of the greatest interest for using possible results over large areas, as well as forecasting yields and profits from olive trees. Our approximation to estimate the biovolume can be very useful to automatically predict the yield and profit in terms of olive production especially if continuous monitoring of biovolume, given that yield per tree data is available. This method can also be extended to monitor tree foliage losses due to disturbances and annual canopy growth, which are useful to assess pruning treatments and for estimating production [EVLS14, JLCT17].

IV. Detection of Spruce Trees (*Picea Abies*) Infected by Bark Beetle in UAV images using YOLOs architectures

1. Introduction

Preserving natural forests is essential for the environment as they play a very important role in the global ecosystem. Unfortunately, there are a number of factors that threaten the well-being of forests. One of them are different pests that can attack trees leading to their weakening or even death. In particular, the European bark beetle (*Ips typographus*, L.) [ADSG18, Werm04] is widespread in the coniferous, mainly spruce forests of Eurasia. It belongs to the class of especially dangerous forest pests. The flight of beetles begins in the spring when the average temperature is about 18 °C. The drying up of Abies trees is accompanied by the mass reproduction of bark beetles that can last 4-5 years, in rare cases up to 9-12 years [ChWB87, Lobi94]. Since the bark beetles cause significant damage to forestry and forest parks, it is important to develop a program for reduction of their quantity. The first step of this program should be forest monitoring. One of the most effective ways of forest conservation monitoring is the use of RS data.

The choice of methods for conducting timely analysis of forest monitoring data is also important. The large size of the forest polygons and the hard-to-reach places under study do not allow the use of pure manual analysis, such as manual counting and individual detection of trees. Since the advent of unmanned aerial vehicles (UAVs) in mass production, it has become possible for scientists to collect local area imagery with very high spatial resolution, but the problem of processing large arrays of information has remained. In particular, the resource-intensive processing of a large amount of unsorted RS data obtained from drones can include an expert opinion on the type of tree, and manual counting of specimens with a particular state of crown integrity. However, recently it was shown that such problems can be effectively solved with the use of deep neural networks [OnIs21, SGMA21, STAR19].

The goal of this work is the detection of infected trees in images obtained from unmanned aerial vehicles (UAVs), by using DL YOLO architectures [BoWL20, RDGF16, ReFa16, ReFa18]. First, from the orthophoto images obtained with UAV, we prepared a dataset for training and testing YOLO architectures. In the next step, we applied a pre-processing procedure to the dataset. This process consists of the enhancement of the contrast of the input image which makes it possible to increase the accuracy of detecting individual tree crowns. Next, we trained and tested the YOLO architectures from 2 to 4 versions. We then presented the results of a comparison of these architectures and identified the best YOLO model for the task of detecting infected trees.

The main contributions of this chapter can be listed as follows:

- We have built a new dataset for the detection of infected trees. The dataset is organized into two subsets of Red-Green-Blue (RGB) color images at 3.75 cm/pixel spatial resolutions. We considered four tree states categories: 1) healthy tree, 2) 25% to 50% damaged tree, 3) 75% damaged tree, and 4) 100% damaged tree.

- We created a new annotator to quickly annotate our dataset.

- We applied pre-processing procedure for improving the quality of the images and hence for increasing the generalization capacity of the detection model.

- We evaluated the object detection YOLO models for the task of infected tree detection in UAV images.

The chapter is organized as follows: The related works are given in Section 2. The materials and methods are presented in Section 3, where the study area is shown in Subsection 3.1, the pre-processing methodology is provided in Subsection 3.2, the annotator of images is described in Subsection 3.3, the YOLO models are evaluated and compared in Subsection 3.4, the experimental setup is described in Subsection 3.5, and the evaluation metric is in Subsection 3.6. The experimental Results are presented in Section 4. The chapter concludes with Section 5.

2. Related works

In this section, we consider related works devoted to the problems of tree detection on ERS data using DL. The object detection is the task of localizing all specified objects of a class and building a bounding box for each of them. Intelligent systems based on NNs can successfully solve problems of plant recognition [BoWL20]. There are other, more traditional approaches to solving these problems, but they do not have the required flexibility outside of limited conditions [OJRJ21, ZTMX17]. NNs provide promising alternative solutions, and many applications benefit from their use.

Below are the works on solving tasks of object detection in the forestry industry on ERS data by DNN, which are the closest to our problem. In the work [XATR20], using a pretrained YOLO model, authors were able to obtain an average accuracy of up to 91.82% in the detection of affected pine trees on UAV-obtained VHR images. Thanks to this solution, it became possible to localize the affected tree on various aerial images. In another work [QGWC20], the authors considered a similar problem of detecting a dead pine tree on UAV data using CNNs of AlexNet and GoogLeNet with a maximum accuracy of up to 97.38%. Tao et al. in [TLZD20] performed the insect-damaged tree detection (dead fir, sick fir, healthy fir, deciduous trees, grass and uncovered) with DJI Mavic 2 pro drone data and DL technique based on CNNs.

In our previous work [STAR19], we were able to detect four categories of fir trees (*Abies Sibirica*) damaged by the *Polygraphus Proximus* Blandford bark beetle in UAV images with DL. However, we solved the problem of detecting individual trees on UAV images using our own algorithm, and then we classified the detected patches as belonging to one of the four categories of tree: a—completely healthy tree or recently attacked by beetles, b—tree colonized by beetles, c—recently died tree, and d—deadwood. In the last step, we proposed a new CNN architecture specially designed to solve that problem. We also presented a comparison of the developed architecture with such models as VGG, ResNet, Inception-V3, InceptionResNet-V2, Xception and DenseNet. It is important to note that our model, trained with data augmentation, showed up to 98.77% accuracy for the next categories of fir trees: completely healthy tree, recently dead tree, and deadwood.

In this chapter, we analyze the detection of four categories of spruce tree state (healthy trees, 25% to 50% damaged trees, 75% damaged trees, and 100% damaged trees) damaged by the European beetle attack on UAV images using different YOLO architectures. Then we compare the quality of the trained architectures on the test data set.

3. Materials and methods

3.1. Study Area

The study area is the territory of the West Balkan Range, the south of the administrative center of Chuprene, Vidin Province, Bulgaria (Figure 4.1).

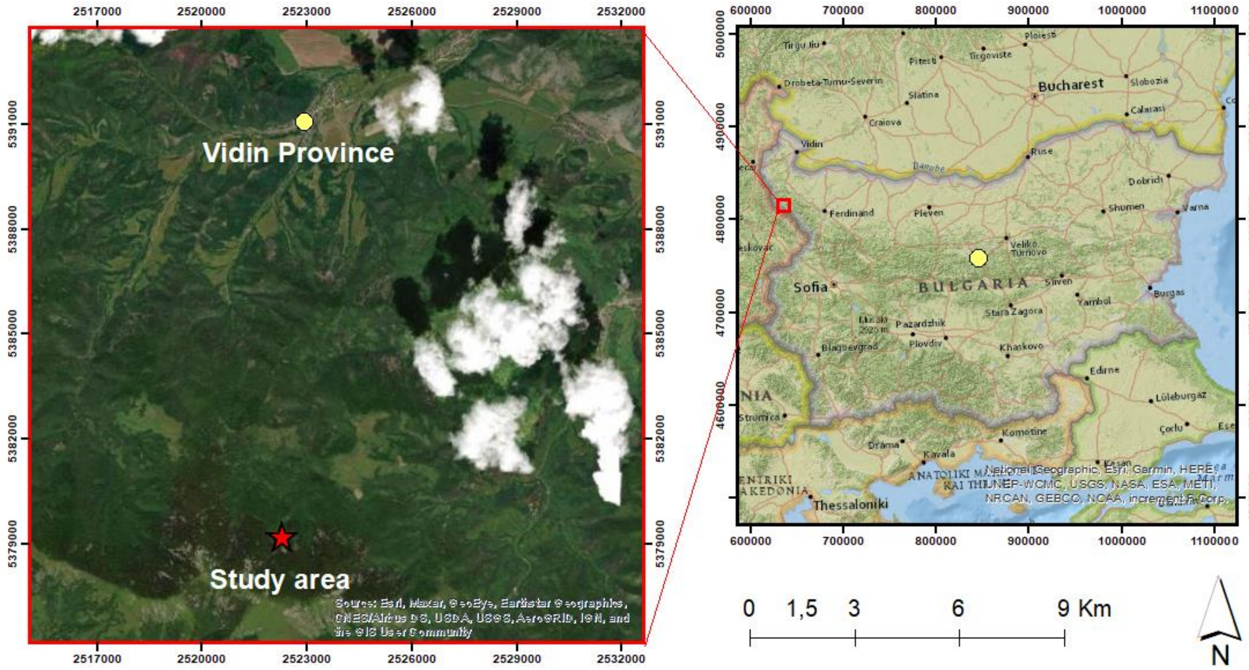

Vidin Province
Study area
Source: Esri, Maxar, GeoEye, Earthstar Geographics, CNES/Airbus DS, USDA, USGS, AeroGRID, IGN, and the GIS User Community
2517000
2520000
2523000
2526000
2529000
2532000
5391000
5388000
5385000
5382000
5379000
600000
700000
800000
900000
1000000
1100000
5000000
4900000
4800000
4700000
4600000
4500000
Bucharest
Sofia
BULGARIA
Thessaloniki
Vidin
National Geographic, Esri, Garmin, HERE, UNEP-WCMC, USGS, NASA, ESA, METI, NRCAN, GEBCO, NOAA, increment P Corp.
0
1,5
3
6
9 Km
N

Figure 4.1. The study area – West Balkan Range, Chuprene, Vidin Province, Bulgaria.

We used ERS obtained from a DJI-Phantom 4 Pro UAV with an RGB camera (Red, Green, and Blue channels) with a resolution of 7 cm/pixel.

The drone flew several times on August 16, 2017, and September 25, 2017, at an altitude of 120 meters above ground. The object of the study is natural forests damaged as a result of attacks by European spruce bark beetles (*Ips typographus*, L.) [TrDL20]. Forests mainly consist of spruce (*Picea Abies*), European beech (*Fagus sylvatica*), Scotch pine (*Pinus sylvestris*), and black pine (*Pinus nigra*). However, in our experiments, the object of research predominantly is damaged spruce trees (*Picea Abies*) in categories as healthy trees, 25% to 50% damaged trees, 75% damaged trees, 100% damaged trees.

We built a dataset made of 400 images, where 80% were for training, 20% for validation, and two plots A and B were for external testing (Table 4.1).

Table 4.1. A brief description of the number of image patches and segments in the dataset of the infected spruce trees, where a—healthy trees, b—25% to 50% damaged trees, c—75% damaged trees, and d—100% damaged trees.

# of Training Images	# of Training Trees				# of Validating Images	# of Validating Trees				Total of Images	Total of Trees
	a	b	c	d		a	b	c	d		
320	312	622	76	188	80	202	400	20	61	400	1881

We pre-processed the original dataset using graphic transformation—pixel contrast enhancement. During the training process, we applied dynamic data augmentation using standard functions: rotation, horizontal flip, vertical flip, resizing. It should also be noted that the number of damaged trees is different as in the experiment we used data from the natural forest. Thus, the total number of trees obtained from one dataset by class is as follows: a—healthy trees (594 trees), b—25% to 50% damaged trees (1206 trees), c—75% damaged trees (104 trees), and d—100% damaged trees (277 trees). The resulting dataset is unbalanced as these are trees from natural forests.

The total number of trees in the test plot A is 38, where 12, 23, 1, and 2 correspond to categories a, b, c, and d, respectively. The total number of trees in the test plot B is 51, where 12, 34, 3, and 2 correspond to categories a, b, c, and d, respectively.

3.2. Pre-processing methodology

UAVs are an affordable and widely used resource for local monitoring of plants. However, in practice, the quality of the images obtained from the UAV can be deteriorated due to the suspended particles in the atmosphere, weather conditions, and the quality of the equipment itself. This results in low/poor contrast images [ChBa88]. In this work, to improve the quality of the images, we used a balance contrast enhancement technique (BCET) [Guo91].

The contrast of the image can be stretched or compressed without changing the histogram pattern of the input image. For this we can use a parabolic function in the general form whose coefficients depend on the input image:

$$I_{New} = a \times (I_{old} - b)^2 + c. \tag{4.1}$$

where I_{old} is the intensity of the input image (of one of the three-color channels), and I_{New} is that of the output. The coefficients a (4.2), b (4.3), and c (4.4) are calculated using the minimum and maximum intensity values of the input (l and h) and output (L and H) image and the intensity mean value of the input (e) and output (E) image:

$$a = \frac{H-L}{(h-l)(h+l-2b)}, \tag{4.2}$$

$$b = \frac{h^2 \cdot (E-L) - s \cdot (H-L) + l^2 \cdot (H-E)}{2 \cdot [h \cdot (E-L) - e \cdot (H-L) + l \cdot (H-E)]}, \tag{4.3}$$

$$c = L - a(l-b)^2, \tag{4.4}$$

where s denotes the intensity mean square sum of the input image (4.5):

$$s = \frac{1}{N} \sum_{i=1}^{N} I_{Old}^2(i), \tag{4.5}$$

where the summation is taken over all image pixels with total number of N. Note that the target values of parameters L, H, and E are set manually.

The example of the quality improvement for one image is shown in Figure 4.2. A clearer visual separation of the individual tree crowns (area of growing and non-growing trees) can be observed in the processed image and in the histogram. Improving local contrast increases the balance of the mean value of the original image based on light and dark edges. This methodology was applied to the original RGB image. The local mean, local standard deviation, minimum and maximum value of the entire image are used to statistically describe the digital image.

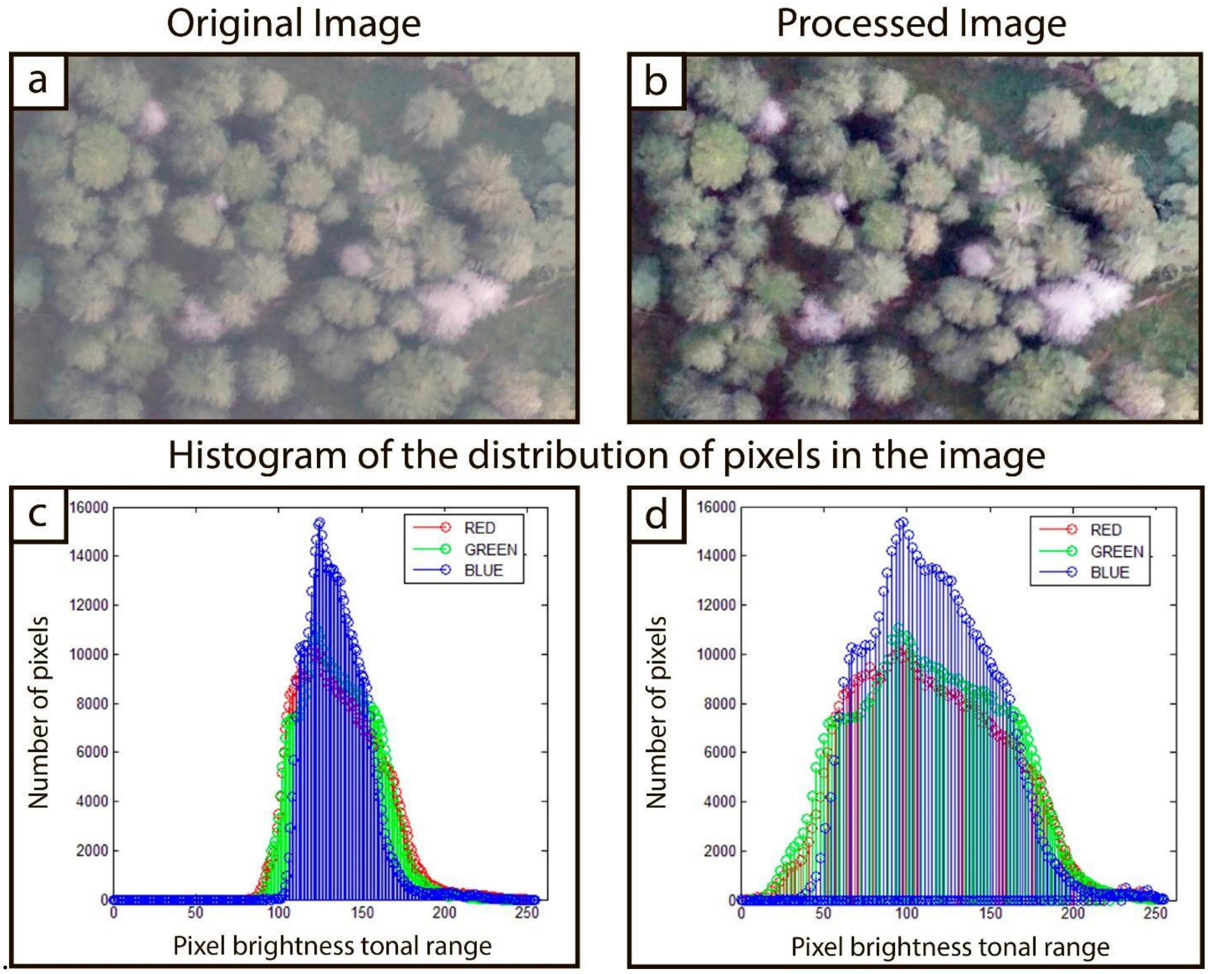


Figure 4.2. Example of pre-processing, where (a) the original image, (b) result of balance contrast enhancement; histogram of input (c) and output (d) image.

Figure 4.2 shows the histogram of the processed image. It indicates that the mean values of all the channels (Red, Green, and Blue) are balanced from 0 to 255, while the mean values of the channels of the source image are from ~90 to ~200. Table 4.2 shows the mean values of the red, green, and blue channels are different from the input and output image of Figure 4.2.

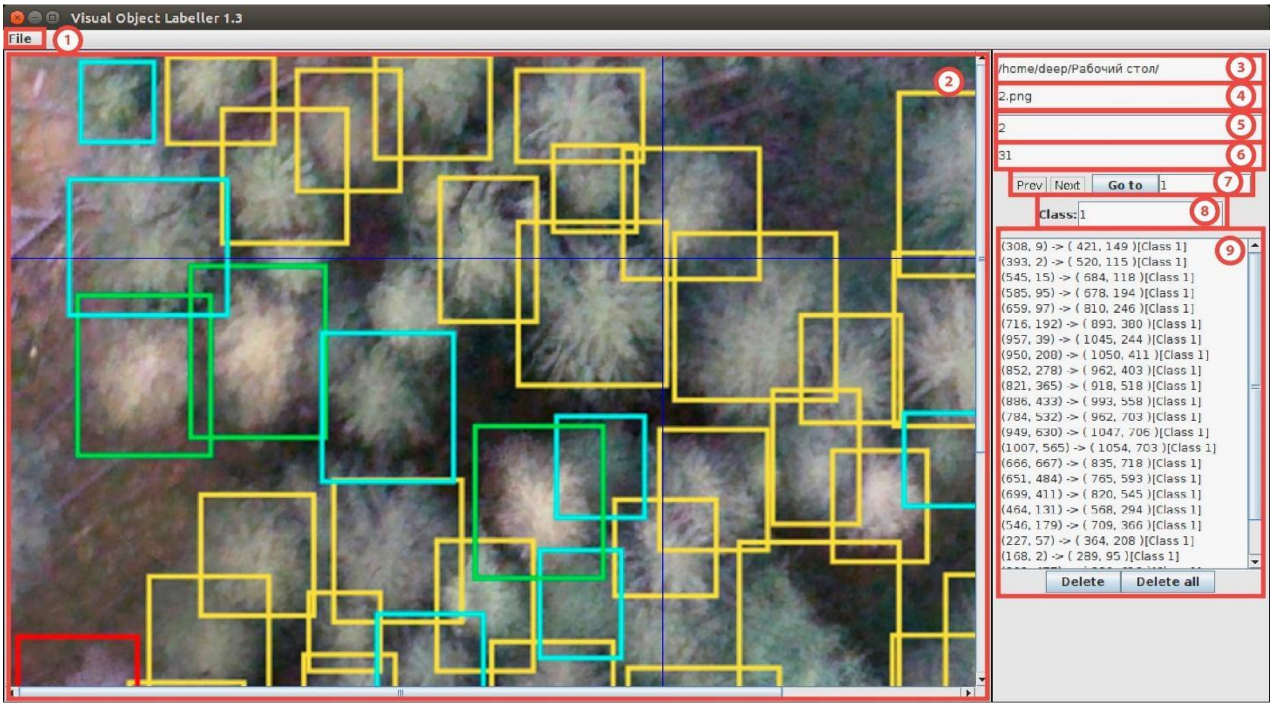
Table 4.2. Statistics of values of the red, green, and blue channels.

Channel	Min values	Max values	Mean value	
	Input Image		Input Image	Output Image
Red	85	210	136.43	119.96
Green	90	205	136.13	119.99
Blue	102	195	135.32	119.95

This kind of color image enhancement takes 0.22 seconds of elapsed time on the image of size 1129×1129 pixels. This process enhances the quality of the image allowing to get better and more accurate results. A histogram value of the red, green, and blue channels after applying a contrast balance enhancement shows that the data has been spread over much more of the available dynamic range. Thus, as a result of the application of preprocessing, the crowns of damaged trees are more clearly distinguished in the image.

3.3. An annotator of dataset

The annotators available include a wide range of functions that allow to label data for any task, which in turn takes time to learn the application and mark up the data. Therefore, for the convenience and acceleration of data annotation for this work, a new lightweight version of the software for annotation according to the YOLO standard was developed to detect objects in images [SMAD18]. The proposed annotator does not require additional graphic libraries and has a convenient interface allowing for fast annotation of large number of images in a few steps. The graphical interface of a software called Visual Object Labeller 1.3 is shown in Figure 4.3.



/home/deep/Рабочий стол/

2.png

2

31

Go to 1

Class: 1

```
(308, 9) -> ( 421, 149 )[Class 1]
(393, 2) -> ( 520, 115 )[Class 1]
(545, 15) -> ( 684, 118 )[Class 1]
(585, 95) -> ( 678, 194 )[Class 1]
(659, 97) -> ( 810, 246 )[Class 1]
(716, 192) -> ( 893, 380 )[Class 1]
(957, 39) -> ( 1045, 244 )[Class 1]
(950, 208) -> ( 1050, 411 )[Class 1]
(852, 278) -> ( 962, 403 )[Class 1]
(821, 365) -> ( 918, 518 )[Class 1]
(886, 433) -> ( 993, 558 )[Class 1]
(784, 532) -> ( 962, 703 )[Class 1]
(949, 630) -> ( 1047, 706 )[Class 1]
(1007, 565) -> ( 1054, 703 )[Class 1]
(666, 667) -> ( 835, 718 )[Class 1]
(651, 484) -> ( 765, 593 )[Class 1]
(699, 411) -> ( 820, 545 )[Class 1]
(464, 131) -> ( 568, 294 )[Class 1]
(546, 179) -> ( 709, 366 )[Class 1]
(227, 57) -> ( 364, 208 )[Class 1]
(168, 2) -> ( 289, 95 )[Class 1]
```

Figure 4.3. Graphical user interface of the Visual Object Labeller version 1.3 annotator with main menu—1, work window—2, file location—3, name and file type—4, selected file—5, total number of files—6, navigation between files—7, assigned class—8, list of mark points—9.

Our version does not require the OpenCV library installed and uses only the standard graphics functions of the JAVA programming language. The basic steps for using the Visual Object Labeller 1.3 to annotate images are presented in Figure 4.4.

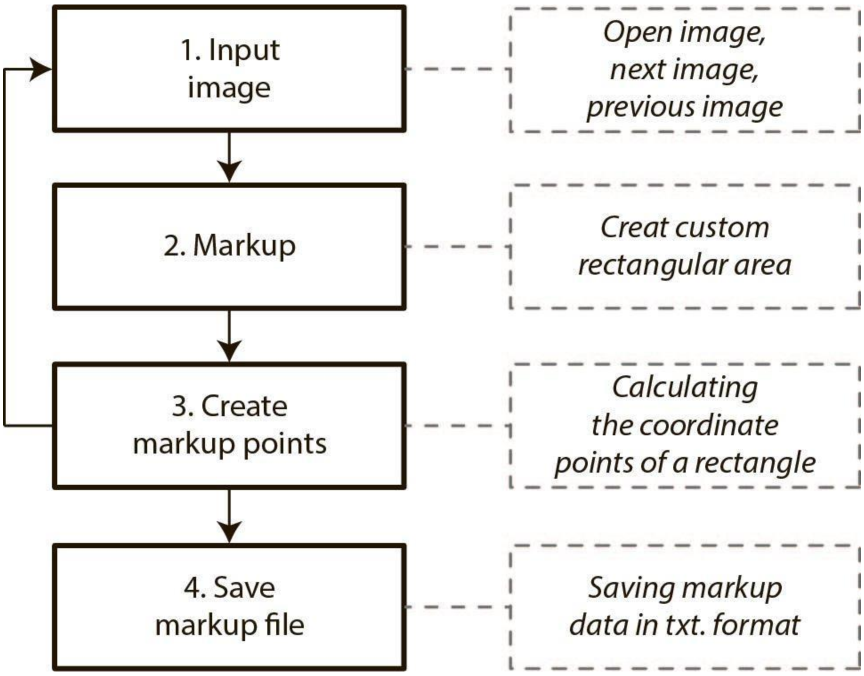

1. Input image
2. Markup
3. Create markup points
4. Save markup file
Open image, next image, previous image
Creat custom rectangular area
Calculating the coordinate points of a rectangle
Saving markup data in txt. format

Figure 4.4. Scheme of operation of the Visual Object Labeller 1.3 annotator.

To work with our own dataset, we had to create markup for the standard view of YOLO models, which it accepts during training and outputs as a result. The text markup of the areas in the image looks like this (4.6):

$$< class >< X_{YOLO} >< Y_{YOLO} >< W_{YOLO} >< H_{YOLO} >, \tag{4.6}$$

where <class> is an integer class number corresponding to infected trees categories a, b, c, and d, respectively. <X_{YOLO}>, <Y_{YOLO}>, <W_{YOLO}>, and <H_{YOLO}> are float values relative to width and height of image in the range (0.0 to 1.0], where X_{box} is the coordinates of the point in the upper-left corner of the rectangle, Y_{box} is the coordinates of the point in the lower-right corner of the rectangle, W_{box} is width of the rectangle, and H_{box} is height of the rectangle, W_{image} is width of the image, H_{image} is height of the image (4.7-4.10):

$$X_{YOLO} = \frac{X_{box} + W_{box} * 0.5}{W_{image}}, \tag{4.7}$$

$$Y_{YOLO} = \frac{Y_{box} + H_{box} * 0.5}{H_{image}}, \tag{4.8}$$

$$W_{YOLO} = \frac{W_{box}}{W_{image}}, \tag{4.9}$$

$$H_{YOLO} = \frac{H_{box}}{H_{image}}. \tag{4.10}$$

3.4. YOLO architectures

Object detection is a fundamental computer vision task in which the algorithm analyzes the input image and outputs a label together with a bounding box that delimits where the object-class is in the image [SGMA21]. In this work we used three YOLO architectures, namely YOLOv2 [ReFa16], YOLOv3 [ReFa18], and YOLOv4 [BoWL20], to compare them and analyze the impact of image preprocessing on the detection. It should be noted that the first version of the YOLO architecture was not included in the comparison, since the first version contains a number of errors and cannot detect small images which we used in the experiment [RDGF16].

Most object detection algorithms take in and process the image multiple times to be able to detect all the objects present in the images. But YOLO looks at the object once. It applies a single forward pass to the whole image and predicts the bounding boxes and their class probabilities. The architecture consists of two major components: feature extractor and feature detector or multi-scale detector. The image is first given to the feature extractor which extracts feature embeddings and then is passed on to the feature detector part of the network that produces the processed image with bounding boxes around the detected classes.

If we compare the selected architectures, then YOLOv1 with the Darknet-19 NN results in more localization errors but is much less likely to predict false positives when searched objects do not present in the data. It outperforms all other detection methods, including deformable part models (DPM) and Regions-based CNN (R-CNN). However, despite the improvements in the YOLOv2 version on Darknet-30 networks, it has better results of detection, but also has a problem while detecting small objects due to down sampling the input image and losing fine-grained features. The improved architecture of YOLOv3 with Darknet-53 and ResNet networks due to its complexity is a bit slower compared to YOLOv2, but at the same time, gives results with higher accuracy. The improved YOLOv4 architecture, in contrast to the previous version, works much faster without loss of definition quality.

3.5. Experimental setup

All models were trained and tested on an Ubuntu 16.04.6 LTS operating system with an NVIDIA GeForce GTX 1060 graphics processing unit (GPU) and CUDA 10.1 parallel computing platform. The input images were reduced to 416×416 pixels to fit the input layer of the training model. We used a global learning rate of 0.001, and four classes have 8000 iterations for maximum batches. Standardized augmentation techniques were used during training where decay is 0.0005, saturation is 1.5, exposure is 1.5.

The loss function [RDGF16] for each of the YOLO models was calculated as follows (4.11):

$$\begin{aligned} &\lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} \left[\left(x_i - \hat{x}_j\right)^2 + \left(y_i - \hat{y}_j\right)^2 \right] \\ &+ \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} \left[\left(\sqrt{w_i} - \sqrt{\widehat{w}_j}\right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{y}_j}\right)^2 \right] \end{aligned} \tag{4.11}$$

$$+\sum_{i=0}^{S^2}\sum_{j=0}^{B}1_{ij}^{obj}\left(C_i-\hat{C}_j\right)^2+\lambda_{noobj}\sum_{i=0}^{S^2}\sum_{j=0}^{B}1_{ij}^{noobj}\left(C_i-\hat{C}_j\right)^2$$
$$+\sum_{i=0}^{S^2}1_i^{obj}\sum_{c\in classes}^{B}\left(p_i(c)-\hat{p}_j(c)\right)^2, \quad (4.11)$$

where S^2 is the output feature map of all grid cells, *B* is the number of bounding box for each grid, *i* is the *i*-th grid, *j* is the *j*-th predicted box of this grid, x_i and y_i are the location of the centroid of the anchor box, w_i and h_i are the width and height of the anchor the box, C_i is the *Objectness* which means a measure of the probability that an object exists in a proposed region of interest, $p_i(c)$ is the probability of real box class, $\hat{p}_j(c)$ is the probability of predict box class, 1_{ij}^{obj} denotes if object appears in cell *i*, $\left(1_{ij}^{obj}, 1_{ij}^{noobj}\right)$ defines whether the *j*-th box in the *i*-th grid is responsible for that prediction, and $\left(\lambda_{coord}, \lambda_{noobj}\right)$ are weighting factors.

The training time for each model was: YOLOv2 - 3 hours, YOLOv3 - 12 hours, YOLOv4 - 6 hours on original images, and YOLOv2 - 4 hours, YOLOv3 - 15 hours, YOLOv4 - 5 hours on pre-processed images.

For detection of infected spruce trees, we trained models on two datasets: without and with data augmentation. The trained models were independently tested on two test plots A and B.

3.6. Evaluation metrics

To evaluate the performance of the trained YOLO architectures in the task of detection of infected spruce trees, we used the mean average precision (mAP) and intersection over union (IoU) metrics. These are popular metrics in measuring the accuracy of object detectors. IoU is a simple scoring metric that requires the following data to apply:

1. The ground-truth labeled bounding boxes, which are created manually.

2. The predicted bounding boxes from the trained model (4.12).

$$IoU = \frac{Area\ of\ Overlap}{Area\ of\ Union}, \quad (4.12)$$

where *overlap* and *union* operations are applied to the ground-truth and corresponding predicted areas.

IoU is used in the mAP metric to measure the overlap of the two boundaries between ground-truth labels and predicted labels from the trained model. If the detection is absolutely correct, then the indicator is equal to 1. The lower the value of the IoU metric, the worse the prediction result. Usually, the threshold of this indicator is set equal to 0.5, meaning that if the IoU > 0.5, then the prediction considered as correct (true) and false otherwise. Based on this, calculations are made of indicators such as Precision (4.13) and Recall (4.14):

$$Precision = \frac{TP}{TP+FP}, \quad (4.13)$$

$$Recall = \frac{TP}{TP+FN}. \tag{4.14}$$

where TP is a true positive prediction, FP is a false positive prediction, and FN is a false negative prediction.

Precision determines the percentage of correctly recognized labels and Recall shows how good true positives were predicted. On these criteria, F1-score and mAP are calculated to evaluate the performance of a model. F1-score is calculated from the precision and recall of the test (4.15). mAP is the average precision (the area under the Precision-Recall curve) averaged over all classes [Boch21]. mAP is calculated in the range from 0 to 1 with the following formula (4.16):

$$F1\,Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}, \tag{4.15}$$

$$mAP = \sum_{i=1}^{N} AP_i = \frac{1}{N} \sum_{Recall_i} Precision(Recall_i), \tag{4.16}$$

where N is the number of classes.

4. Experimental results

In this section, we present the results of our experiments. The purpose of this experiment was testing and comparing YOLOv2 [ReFa16], YOLOv3 [ReFa18] and YOLOv4 [BoWL20] architectures on images taken from UAV. The main task is detecting infected trees on four categories as follows: a—healthy trees, b—25% to 50% damaged trees, c—75% damaged trees, and d—100% damaged trees. The models were trained on the original images and pre-processed images, with data-augmentation, and tested on two test plots, A and B.

The results from training all YOLO architectures are presented in Table 4.3.

Table 4.3. Results from training YOLO architectures on two datasets.

Architecture	Loss function		Training epochs	
	Original dataset	**Pre-processed dataset**	**Original dataset**	**Pre-processed dataset**
YOLOv2	0.28	0.33	4500	2000
YOLOv3	1.57	0.8	6500	6500
YOLOv4	0.8	0.93	8000	8000

We used two pre-processed test plots A and B to test the trained YOLO architectures. The total number of trees in test plot A was 38, where 12, 23, 1, and 2 correspond to categories a, b, c, and d, respectively. The total number of trees in test plot B was 51, where 12, 34, 3, and 2 correspond to categories a, b, c, and d, respectively. Table 4.4 shows the results of the predictions of YOLO architectures for each infected tree of the categories, presented in relation to the samples labeled by an expert.

Table 4.4. Prediction results of YOLO architectures for each infected tree of the categories, where a—healthy trees, b—25% to 50% damaged trees, c—75% damaged trees, and d—100% damaged trees.

	Test plot A				**Test plot B**				**Test plot A**				**Test plot B**			
	Original image								**Pre-processed image**							
	a	**b**	**c**	**d**	**a**	**b**	**c**	**d**	**a**	**b**	**c**	**d**	**a**	**b**	**c**	**d**
YOLOv2																
Ground truth	12	24	1	2	10	41	4	4	12	24	1	2	10	41	4	4
Predicted	**15**	37	**1**	**2**	16	53	**4**	0	17	27	**1**	3	18	49	**3**	1
YOLOv3																
Ground truth	12	24	1	2	10	41	4	4	12	24	1	2	10	41	4	4
Predicted	9	**19**	**1**	**2**	5	15	3	0	16	13	**1**	**2**	4	**40**	0	**3**
YOLOv4																
Ground truth	12	24	1	2	10	41	4	4	12	24	1	2	10	41	4	4
Predicted	20	12	2	**2**	**12**	**39**	3	0	8	**20**	**1**	3	17	39	1	**3**

It can be noted that the first and second categories of infected trees are best detected by the YOLOv4 architecture on both datasets. The best detection of the third category on data with and without pre-processing is provided by the YOLOv2 architecture. The fourth category on the original data is better detected by the YOLOv3 architecture, and on data with pre-processing it is better detected by the YOLOv3 and YOLOv4 architectures. In general, YOLOv4 showed the best result in detecting infected trees, as it showed fewer false positives and more correct answers. A graphical presentation of the results of Table 4.4 is shown in Figure 4.5.

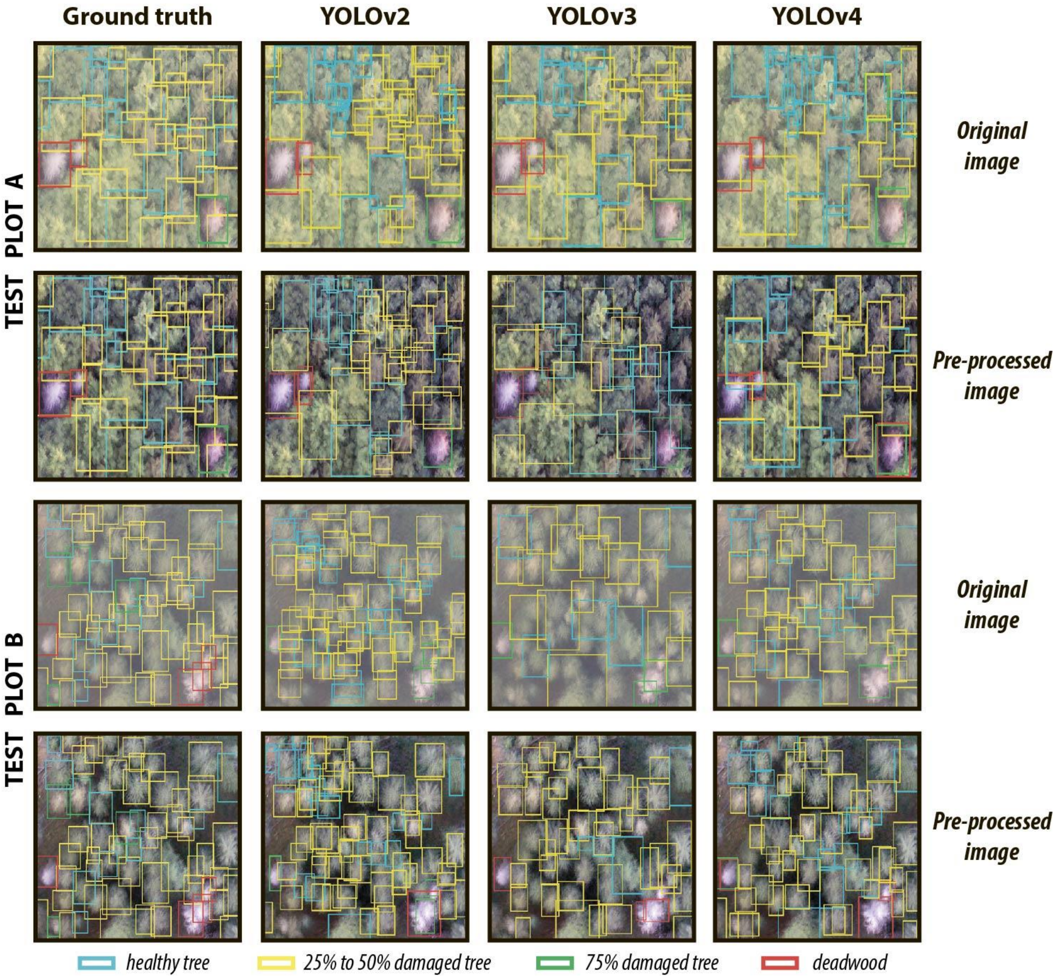


Figure 4.5. Comparative result of detecting infected trees of trained YOLO architectures on two test plots, where Blue is a healthy tree, Yellow is 25% to 50% damaged tree, Green is 75% damaged tree, and Red is 100% damaged tree.

Table 4.5 presents the performance results of testing YOLO architectures calculated according to the metrics presented in Section 3.6.

Table 4.5. Average results of the three YOLO architectures performance on the external A and B plots.

Performance	**YOLOv2**		**YOLOv3**		**YOLOv4**	
	Original dataset	**Pre-processed dataset**	**Original dataset**	**Pre-processed dataset**	**Original dataset**	**Pre-processed dataset**
IoU	0.4	0.76	0.68	0.74	0.71	0.73
Precision	0.7	**0.9**	**0.93**	**0.96**	**0.94**	**0.95**
Recall	0.56	**0.96**	0.48	**0.91**	0.72	0.76
mAP	0.32	**0.92**	**0.9**	**0.97**	**0.91**	**0.94**
Loss	0.62	0.93	0.63	0.93	0.81	0.84

From the results presented in Table 4.5 follows that preprocessed images for training can be used for each of the architectures with the degree of the results improvement depending on the category. Contrast enhancement results in the improvement for YOLOv2 in detection of categories 3 and 4, for YOLOv3 of categories 3 and 4, and for YOLOv4 of categories 1, 2 and 4. In general, after applying contrast enhancement the accuracy of architectures increased significantly and there were fewer false positives and false negatives.

It should be noted that testing each of the test areas took a different amount of time for each version of the YOLOs. For example, the processing of images with a size of 1280×720 pixels using the CUDA 10.1 parallel computing platform and NVIDIA GeForce GTX 1060 GPU on the trained YOLOv2 architecture took about 22.22 microseconds, and for the trained YOLOv3 and YOLOv4 architectures with the same image input parameters it was 31.25 milliseconds and 28.57 milliseconds, respectively.

5. Conclusion

According to the results of the experiments, it can be concluded that for a better result in the task of detecting infected spruce trees damaged by the bark beetle and counting the number of detected specimens in images obtained from UAV cameras, it is preferable to use the YOLOv4 architecture, trained on a dataset pre-processed by increasing the pixel contrast. It should be noted that the dataset composed of the pre-processed images showed a very good mAP metric at all stages of model training. The test result on these images improved by 65% for YOLOv2, by 7.22% for YOLOv3, and by 3.19% for YOLOv4. Also, the results of a preliminary analysis of the values of the image processing speed show that on medium resources (NVIDIA GeForce GTX 1060 GPU) in real time, the architecture of the YOLOv2 version can be used as an expert agent for analyzing local areas using UAVs.

V. Conclusion

1. Particular conclusions

This thesis has addressed three relevant problems from natural sciences, focusing on plant species detection in aerial and satellite images using DL. The trained CNN algorithms developed in this work are capable of performing local detection of individual plant species on UAV images in a relatively short time.

The second chapter presented a DL-methodology that determined the tree damage stage based on the shape, texture, and color of the tree crown in UAV images. The results show that DL methods are outstanding models for this task, and thus we were encouraged to shift towards it as the main technology for the development of the rest of the proposals. Also, out of all possible data, images obtained from UAVs are viable and suitable for solving problems of detection and segmentation of plants. However, with the help of a UAV, it is possible to survey the territory only locally, therefore, not much such data can be obtained due to the fact that the drone cannot fly over all areas of the earth due to the limited fly distances. At the same time, the images have a high spatial resolution, which indicates their suitability for solving a number of RS problems. The preparation of the dataset is usually performed manually and is time consuming, but the experimental results show that the preliminary processing of the ERS data significantly improves the performance evaluation indicators of the developed and existing models of NNs. Also, this work shows that the new NN model can solve the specific problem of detecting damaged trees in four categories using UAV images with higher accuracy than other very powerful existing models. Thus, this work represents an important contribution to the community of researchers, agronomists, and organizations involved in forest monitoring.

The results of this chapter were published:

- **Safonova, A.**; Tabik, S.; Alcaraz-Segura, D.; Rubtsov, A.; Maglinets, Y.; Herrera, F. Detection of Fir Trees (*Abies sibirica*) Damaged by the Bark Beetle in Unmanned Aerial Vehicle Images with Deep Learning. *Remote Sens*. 2019, 11, 643, https://doi.org/10.3390/rs11060643.
  - **Status:** Published.
  - **Impact Factor (JCR 2019):** 4.509
  - **Subject Category:** Remote Sensing
  - **Ranking:** 9/30
  - **Quartile:** Q2
  - **Citations according to google scholar:** 50

The third chapter presented a particular case of semantic segmentation of olive trees to further calculate the biovolume of the tree and a possible prediction of the yield and profit. In particular, we have adapted one of the well-known segmentation models, Mask R-CNN, to solve a relatively new urgent problem of tree segmentation based on RS data. Our results show that trained CNN models on a small new UAV dataset with augmentation reached very high accuracies. This generalization opens the possibility of using multispectral or RGB imagery over extensive areas at a lower cost per hectare for the purpose of tree volume monitoring, with wide implications in precision agriculture, precision forestry, and precision restoration. As shown in the results, the segmentation of tree crown and tree shadow based on CNN can be used to approximate biovolume trees. It can be very useful to automatically predict the yield and profit in terms of olive production especially if continuous monitoring of biovolume, given that yield per tree data is available. This method can also be extended to monitor tree foliage losses due to disturbances and annual canopy growth, which are useful to assess pruning treatments and for estimating production.

The results of this chapter were published:

- **Safonova, A.**; Guirado, E.; Maglinets, Y.; Alcaraz-Segura, D.; Tabik, S. Olive Tree Biovolume from UAV Multi-Resolution Image Segmentation with Mask R-CNN. *Sensors* 2021, 21, 1617, https://doi.org/10.3390/s21051617.
  - **Status:** Published.
  - **Impact Factor (JCR 2019):** 3.275
  - **Subject Category:** Computer Science
  - **Rang:** 15/64 in category Remote Sensing and 70/300 in category Instruments & Instrumentation
  - **Quartile:** Q1 in category Instruments & Instrumentation and Q2 in Remote Sensing

The fourth chapter was focused on solving the problem of object detection by applying a powerful model based on the DL technique. We conducted training and comparison of three YOLO architectures - 2, 3, and 4 on a new small pre-processed dataset, where we found out that these algorithms are suitable for solving problems of detecting trees on RS data. The main solution for this problem is on-time prevention of pests outbreak by identifying damaged trees in their early stages. Also, the results of a preliminary analysis show that the YOLO architectures can be used to help an expert agent for analyzing local areas using UAVs in real time.

The results of this chapter were submitted:

- **Safonova, A.**; Alekhina, A.; Hamad, Y.; Tabik, S. Detection of Spruce Trees (*Picea Abies*) Infected by Bark Beetle in UAV images using YOLOs architectures. *IEEE Transactions on Geoscience and Remote Sensing*.
  - **Status:** Submitted
  - **Impact Factor:** 5.855
  - **Subject Category:** Earth and Planetary Sciences
  - **Rang:** 5/30 in Remote Sensing and 5/27 in Imaging Science & Photographic Technology
  - **Quartile:** Q1 in Imaging Science & Photographic Technology and Q1 in Remote Sensing

2. General conclusions

This thesis presents one of the first studies in exploring the potential of deep CNNs, data preprocessing and RS data, in addressing plant species conservation problems. In particular, this thesis presents the results and analysis of deep CNN models in three different problems from natural sciences:

1. The detection of Fir trees (*Abies Sibirica*) damaged by the bark beetle in UAV images using DL.

2. The estimation of olive tree biovolume from UAV multi-resolution image segmentation using Mask R-CNN.

3. The detection of Spruce trees (*Picea Abies*) infected by bark beetle in UAV images using YOLOs architectures.

The main findings of this thesis are as follows:

- CNN-based detection and segmentation models are robust in fir tree, spruce tree and olive tree species monitoring. In combination with data-augmentation techniques, the developed detection and segmentation models can learn and generalize correctly even with small datasets.

- CNN-based detection and segmentation models are suitable for monitoring different plant species using high resolution multi-band RS data.

- The quality of these models depends strongly on the quality of the data and the correct formulation of the problem. High quality data involves cleaner and higher spatial resolution images labeled by experts of each specific problem.

- An optimal labeling and problem definition must be performed with experts under a close collaboration with data scientists.

- The developed models can be used in the detection and monitoring of other different plant species.

All the proposed models can be applicable to satellite images with spatial resolutions of the order of 0.3 meter per pixel or higher.

The results of the first two chapters of this thesis have been published in two journals ranked as Q1 and Q2 in JCR. The results of the third chapter have been submitted to “IEEE Transactions on Geoscience and Remote Sensing”.

3. Future work

From the conclusions drawn from this thesis, a new and promising line of research can be proposed. Our aim is to improve and adapt existing DL models for addressing new emerging problems in RS and natural sciences.

- **Application of the developed and other new DL methods in solving problems of detection and segmentation of plants based on data from satellite observation of the Earth.** DL techniques have already proved themselves as the best methods to solve problems of RS data [MLZY19]. However, there is a need to adapt methods for high- and medium-resolution satellite data (more than 1 meter per pixel) to allow monitoring sites globally.

- **Pre-processing procedures for ERS images.** Another interesting topic for the development of research in related fields ERS and DL is the research and improvement of image preprocessing techniques. Considering the fact that for NN models trained on data with preliminary processing, the performance indicators are higher than that of the model trained on the original data, the need for the implementation and improvement of preprocessing techniques is evident.

- **Plant mapping based on RS data using DL methods.** Terrain mapping, one of the relevant areas of ERS, also has a significant breakthrough due to the introduction of DL methods. For example, using DL technologies, it is possible to map healthy and damaged forests, forest areas by tree species composition, and even individual trees [OnIs21, SyDB19, TAFG19].

- **3D modeling of maps of plants terrain according to RS data using DL.** 3D modeling of the terrain, and in particular the territories occupied by vegetation, allows one to solve a number of different problems [ICNK17, QiGr21]. In turn, the introduction of DL technology in 3D terrain modeling can improve the accuracy results in comparison with classical methods, which is another interesting direction of development in science.

Appendix A

Table A1. Performance comparison of the training by the proposed CNN model with other models.

Model	Without augmentation		With Augmentation	
	Loss	Accuracy	Loss	Accuracy
Our CNN model	0.05	1	0.001	1
Xception	0.3	0.89	0.002	1
VGG16	0.03	1	0.03	1
VGG19	0.07	1	0.01	0.99
ResNet50 + 4	0.21	0.94	0.01	1
InceptionV3 + 4	0.34	0.89	0.003	1
InceptionResNetV2 + 4	0.13	0.94	0.04	0.99
DenseNet121 + 4	0.17	0.94	0.02	0.99
DenseNet169 + 4	0.08	1	0.004	0.99
DenseNet201 + 4	0.12	0.94	0.01	1

Table A2. The results of testing alternative models for each class that were trained on a data set with augmentation.

Categories of trees	TP	TN	FP	FN	Acc (%)	Precision (%)	Recall (%)	F-score (%)
Xception								
1	30	26	19	2	72.73	61.22	93.75	74.07
2	15	41	8	21	65.88	65.22	41.67	50.85
3	7	49	1	9	**84.85**	87.5	53.75	58.33
4	4	52	4	0	**93.33**	50	100	66.67
VGG16								
1	24	43	3	8	**85.90**	88.89	75	**81.36**
2	33	34	14	3	**79.76**	70.21	91.67	**79.52**
3	6	61	0	10	**87.01**	100	37.5	54.55
4	4	63	4	0	**94.37**	50	100	66.67
VGG19								
1	26	38	6	6	**84.21**	81.25	81.25	**81.25**
2	29	35	11	7	**78.05**	72.5	80.56	**76.32**
3	5	59	0	11	**85.33**	100	31.25	47.62
4	4	60	7	0	**90.14**	36.36	100	53.33
ResNet50								
1	32	32	16	0	**80**	66.67	100	**80**
2	20	44	1	16	**79.01**	95.24	55.56	**70.18**
3	9	55	1	7	**88.89**	90	56.25	69.23
4	3	61	6	1	**90.14**	33.33	75	46.15
InceptionV3								
1	30	25	21	2	70.51	58.82	93.75	72.29
2	16	39	8	20	66.27	66.67	44.44	53.33
3	6	49	1	10	**83.33**	85.71	37.50	52.17
4	3	52	3	1	**93.22**	50	75	60
InceptionResNetV2								
1	23	27	14	9	68.49	62.16	71.88	66.67
2	18	32	11	18	63.29	62.07	50	55.38
3	6	44	6	10	**75.76**	50	37.5	42.86
4	3	47	7	1	**86.21**	30	75	42.86
DenseNet121								
1	26	33	10	6	**78.67**	72.22	81.25	**76.47**
2	24	35	8	12	74.68	75	66.67	70.59

Continuation of Table A2. The results of testing alternative models for each class that were trained on a data set with augmentation.

Categories of trees	TP	TN	FP	FN	Acc (%)	Precision (%)	Recall (%)	F-score (%)
3	6	53	1	10	**84.29**	85.71	37.50	52.17
4	3	56	10	1	**84.29**	23.08	75	35.29
DenseNet169								
1	32	26	21	0	73.42	60.38	100	**75.29**
2	15	43	5	21	69.05	75	41.67	53.57
3	8	50	1	8	**86.57**	88.89	50	64
4	3	55	3	1	**93.55**	50	75	60
Dense Net201								
1	32	22	22	0	71.05	59.26	100	74.42
2	15	39	7	21	65.85	68.18	41.67	51.72
3	4	50	1	12	**80.6**	80	25	38.1
4	3	51	4	1	**91.53**	42.86	75	54.55

Bibliography

[ADSG18] Abdullah H., Darvishzadeh R., Skidmore A.K., Groen, T.A., Heurich M. European spruce bark beetle (*Ips typographus*, L.) green attack affects foliar reflectance and biochemical properties. In: *International Journal of Applied Earth Observation and Geoinformation* Bd. 64 (2018), S. 199–209.

[BaAA11] Baranchikov Y., Akulov E., Astapenko S. Bark beetle Polygraphus Proximus: a new aggressive far eastern invader on Abies species in Siberia and European Russia. In: *Proceedings.* 21st U.S. Department of Agriculture, Forest Service, Northern Research Station: 64-65. Bd. p-75 (2011), S. 64–65.

[BaHa91] Barlow J.F., Harrison G. Shaded by Trees? In: *Trees in focus. Practical Care and Management, Arboricultural Practice Notes*. (1991), S. 1–8.

[BDMC19] Brito, C., Dinis L-T., Moutinho-Pereira J., Correia C.M. Drought Stress Effects and Olive Tree Acclimation under a Changing Climate. In: *Plants* Bd. 8 (2019), Nr. 7.

[BNMA17] Baeta R., Nogueira K., Menotti D., dos Santos, J.A. Learning Deep Features on Multiple Scales for Coffee Crop Recognition. In: *2017 30th SIBGRAPI Conference on Graphics, Patterns and Images*. 2017, S. 262–268.

[Boch21] Bochkovskiy A. Yolo v4, v3 and v2 for Windows and Linux, 2021. URL: https://github.com/AlexeyAB.

[Bona08] Bonan G.B. Forests and Climate Change: Forcings, Feedbacks, and the Climate Benefits of Forests. In: *Science Bd.* 320, American Association for the Advancement of Science (2008), Nr. 5882, S. 1444–1449.

[BoWL20] Bochkovskiy A., Wang C-Y., Liao H-Y. M. YOLOv4: Optimal Speed and Accuracy of Object Detection. In: *arXiv:2004.10934* [cs, eess] (2020).

[BSVV19] Basso M., Stocchero D., Ventura Bayan Henriques R., Vian Al., Bredemeier C., Konzen AA. Pignaton de Freitas E. Proposal for an Embedded System Architecture Using a GNDVI Algorithm to Support UAV-Based Agrochemical Spraying. In: *Sensors* Bd. 19 (2019), Nr. 24.

[BTKR20] Brandt M., Tucker C.J., Kariryaa A., Rasmussen K., Abel C., Small J., Chave J., Rasmussen L.V. An unexpectedly large count of trees in the West African Sahara and Sahel. In: *Nature* Bd. 587, Nature Publishing Group (2020), Nr. 7832, S. 78–82.

[Calc19] Calculus III - Green’s Theorem. URL: tutorial.math.lamar.edu/Classes/CalcIII/GreensTheorem.aspx.

[CBMO19] Castelão T., Everton B.M., Bruno M., Gabriel K., Oliveira A. da S., Alvarez M., Amorim W.P., De Souza Belete N.A., da Silva G.G. U.A.: Automatic Recognition of Soybean Leaf Diseases Using UAV Images and Deep Convolutional Neural Networks. In: *IEEE Geoscience and Remote Sensing Letters* (2019), S. 1–5.

[CCJL18] Csillik O., Cherbini J., Johnson R., Lyons A., Kelly M. Identification of Citrus Trees from Unmanned Aerial Vehicle Imagery Using Convolutional Neural Networks. In: *Drones* Bd. 2, Multidisciplinary Digital Publishing Institute (2018), Nr. 4, S. 39.

[ChBa88] Christiansen E., Bakke. ALF: The Spruce Bark Beetle of Eurasia. In: BERRYMAN, A. A. (Hrsg.): Dynamics of Forest Insect Populations: Patterns, Causes, Implications, Population Ecology. Boston, MA: *Springer* US, 1988 — ISBN 978-1-4899-0789-9, S. 479–503.

[ChWB87] Christiansen E., Waring R.H., Berryman A.A. Resistance of conifers to bark beetle attack: Searching for general relationships. In: *Forest Ecology and Management* Bd. 22 (1987), Nr. 1, S. 89–106.

[CiMS12] Ciregan D., Meier U., Schmidhuber J. Multi-column deep neural networks for image classification. In: *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012, S. 3642–3649.

[CLGJ17] Chen Y., Li C., Ghamisi P., Jia X., Gu Y. Deep Fusion of Remote Sensing Data for Accurate Classification. In: *IEEE Geoscience and Remote Sensing Letters* Bd. 14 (2017), Nr. 8, S. 1253–1257.

[Coco21] COCO - Common Objects in Context. URL: cocodataset.org/#keypoints-eval.

[Comm18] Jeremy J. Common architectures in convolutional neural networks. URL: jeremyjordan.me/convnet-architectures/.

[CRDD15] Candiago S., Remondino F., de Giglio M., Dubbini M., Gattelli M. Evaluating Multispectral Images and Vegetation Indices for Precision Farming Applications from UAV Images. In: *Remote Sensing* Bd. 7, Multidisciplinary Digital Publishing Institute (2015), Nr. 4, S. 4026–4047.

[CVVG18] Cárdenas D.A.G., Valencia J.A.R., Velásquez D.F.A., Gonzalez J.R.P. Dynamics of the Indices NDVI and GNDVI in a Rice Growing in Its Reproduction Phase from Multi-spectral Aerial Images Taken by Drones. In: *Advances in Information and Communication Technologies for Adapting Agriculture to Climate Change II*: Springer, Cham, 2018, S. 106–119.

[DeBY16] Deli Z., Bingqi C., Yunong Y. Farmland Scene Classification Based on Convolutional Neural Network. In: *2016 International Conference on Cyberworlds* (CW), 2016, S. 159–162.

[DSMB20] Delancey E.R., Simms J.F., Mahdianpari M., Brisco B., Mahoney C., Kariyeva J. Comparing Deep Learning and Shallow Learning for Large-Scale Wetland Classification in Alberta, Canada. In: *Remote Sensing* Bd. 12, Multidisciplinary Digital Publishing Institute (2020), Nr. 1, S. 2.

[DWPH17] Dash J.P., Watt M.S., Pearse G.D., Heaphy M., Dungey H.S. Assessing very high resolution UAV imagery for monitoring forest health during a simulated disease outbreak. In: *ISPRS Journal of Photogrammetry and Remote Sensing* Bd. 131 (2017), S. 1–14.

[DyKM16] Dyrmann M., Karstoft H., Midtiby H. Plant species classification using deep convolutional neural network. In: *Biosystems Engineering* Bd. 151 (2016), S. 72–80.

[EVLS14] Estornell Cremades J., Velázquez Martí B., López Cortés I., Salazar Hernández D.M., Fernández-Sarría A. Estimation of wood volume and height of olive tree plantations using airborne discrete-return LiDAR data. In: *GIScience and Remote Sensing* Bd. 51: Taylor &amp;amp; Francis: STM, Behavioural Science and Public Health Titles, 2014, S. 17–29.

[FLGX18] Fan ZH., Lu J., Gong M., Xie H., Goodman E.D. Automatic Tobacco Plant Detection in UAV Images via Deep Neural Networks. In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* Bd. 11 (2018), Nr. 3, S. 876–887.

[FuMI83] Fukushima K., Miyake S., Ito T. Neocognitron: A neural network model for a mechanism of visual pattern recognition. In: *IEEE Transactions on Systems,* Man, and Cybernetics Bd. SMC-13 (1983), Nr. 5, S. 826–834.

[GACP20] Guirado E., Alcaraz-Segura D., Cabello J., Puertas-Ruíz S., Herrera F., Tabik S. Tree Cover Estimation in Global Drylands from Space Using Deep Learning. In: *Remote Sensing* Bd. 12, Multidisciplinary Digital Publishing Institute (2020), Nr. 3, S. 343.

[GBRT21] Guirado E., Blanco-Sacristán J., Rodríguez-Caballero E., Tabik S., Alcaraz-Segura D., Martínez-Valderrama J., Cabello J. Mask R-CNN and OBIA Fusion Improves the Segmentation of Scattered Vegetation in Very High-Resolution Optical Sensors. In: *Sensors* Bd. 21, Multidisciplinary Digital Publishing Institute (2021), Nr. 1, S. 320.

[GDDM14] Girshick R., Donahue J., Darrell T., Malik J. Rich feature hierarchies for accurate object detection and semantic segmentation. In: *arXiv:1311.2524* [cs] (2014).

[GIEM19] Gonzalez-Fernandez I., Iglesias-Otero M.A., Esteki M., Moldes O.A., Mejuto J.C., Simal-Gandara J. A critical review on the use of artificial neural networks in olive oil production, characterization and authentication. In: *Critical Reviews in Food Science and Nutrition* Bd. 59 (2019), Nr. 12, S. 1913–1926.

[GiKM96] Gitelson A.A., Kaufman Y.J., Merzlyak M.N. Use of a green channel in remote sensing of global vegetation from EOS-MODIS. In: *Remote Sensing of Environment* Bd. 58 (1996), Nr. 3, S. 289–298.

[Girs15] Girshick R. Fast R-CNN. In: *arXiv:1504.08083* [cs] (2015).

[GTAC17] Guirado E., Tabik S., Alcaraz-Segura D., Cabello J., Herrera F. Deep-learning Versus OBIA for Scattered Shrub Detection with Google Earth Imagery: Ziziphus lotus as Case Study. In: *Remote Sensing* Bd. 9 (2017), Nr. 12, S. 1220.

[GuKN19] Gurumurthy V.A., Kestur R., Narasipura O. Mango Tree Net - A fully convolutional network for semantic segmentation and individual crown detection of mango trees. In: *arXiv:1907.06915* [cs] (2019).

[Guo91] GUO L.J. Balance contrast enhancement technique and its application in image colour composition. In: *International Journal of Remote Sensing* Bd. 12, Taylor & Francis (1991), Nr. 10, S. 2133–2151.

[HGDG18] He K., Gkioxari G., Dollár P., Girshick R. Mask R-CNN. In: *arXiv:1703.06870* [cs] (2018).

[HHST18] Hunt E.R., Horneck D.A., Spinelli C.B., Turner R.W., Bruce A.E., Gadler D.J., Brungardt J.J., Hamm P.B. Monitoring nitrogen status of potatoes using small unmanned aerial vehicles. In: *Precision Agriculture* Bd. 19, Springer (2018), Nr. 2, S. 314–333.

[HKGP17] Handique B.K., Khan A.Q., Goswami C., Prashnani M., Gupta C., Raju P.L.N. Crop Discrimination Using Multispectral Sensor Onboard Unmanned Aerial Vehicle. In: *Proceedings of the National Academy of Sciences*, India Section A: Physical Sciences Bd. 87 (2017), Nr. 4, S. 713–719.

[HLMW18] Huang G., Liu Z., Van der Maaten L., Weinberger K.Q. Densely Connected Convolutional Networks. In: *arXiv:1608.06993* [cs] (2018).

[HOAS13] Heurich M., Ochs T., Andresen T., Schneider T. Object-orientated image analysis for the semi-automatic detection of dead trees following a spruce bark beetle (Ips typographus) outbreak. In: *European Journal of Forest Research* Bd. 129 (2013), S. 313–324.

[HoMe18] Holloway J., Mengersen K. Statistical Machine Learning Methods and Remote Sensing for Sustainable Development Goals: A Review. In: *Remote Sensing* Bd. 10, Multidisciplinary Digital Publishing Institute (2018), Nr. 9, S. 1365.

[HPMH13] Hansen M.C., Potapov P.V., Moore R., Hancher M., Turubanova S.A., Tyukavina A., Thau D., Stehman S.V., U.A.: High-Resolution Global Maps of 21st-Century Forest Cover Change. In: *Science* Bd. 342 (2013), Nr. 6160, S. 850–853.

[HSSM19] Hartling S., Sagan V., Sidike P., Maimaitijiang M., Carron J. Urban Tree Species Classification Using a WorldView-2/3 and LiDAR Data Fusion Approach and Deep Learning. In: *Sensors* Bd. 19, Multidisciplinary Digital Publishing Institute (2019), Nr. 6, S. 1284.

[HWKC16] Helbig M., Wischnewski K., Kljun N., Chasmer L.E., Quinton W.L., Detto M., Sonnentag O. Regional atmospheric cooling and wetting effect of permafrost thaw-induced boreal forest loss. In: *Global Change Biology* Bd. 22 (2016), Nr. 12, S. 4048–4066.

[HZRS15] He K., Zhang X., Ren S., Sun J. Deep Residual Learning for Image Recognition. In: *arXiv:1512.03385* [cs] (2015).

[ICNK17] Ioannidou A., Chatzilari E., Nikolopoulos S., Kompatsiaris I. Deep Learning Advances in Computer Vision with 3D Data: A Survey. In: *ACM Computing Surveys* Bd. 50 (2017), Nr. 2, S. 20:1-20:38.

[JLCT17] Jiménez-Brenes F.M., López-Granados F., de Castro A.I., Torres-Sánchez J., Serrano N., Peña J.M. Quantifying pruning impacts on olive tree architecture and annual canopy growth by using UAV-based 3D modelling. In: *Plant Methods* Bd. 13 (2017), Nr. 1, S. 55.

[Kera21] Keras: the Python deep learning API. URL: keras.io.

[Kerc14] Kerchev I. Ecology of four-eyed fir bark beetle Polygraphus proximus Blandford (Coleoptera; Curculionidae, Scolytinae) in the west Siberian region of invasion. In: *Russian Journal of Biological Invasions* Bd. 5 (2014), S. 176–185.

[KLSS17] Kussul N., Lavreniuk M., Skakun S., Shelestov A. Deep Learning Classification of Land Cover and Crop Types Using Remote Sensing Data. In: *IEEE Geoscience and Remote Sensing Letters* Bd. PP (2017), S. 1–5.

[KMGO19] Kitano B.T., Mendes C.C.T., Geus A.R., Oliveira H.C., Souza J.R. Corn Plant Counting Using Deep Learning and UAV Images. In: IEEE *Geoscience and Remote Sensing Letters* (2019), S. 1–5.

[KrSH12] Krizhevsky A., Sutskever I., Hinton G.E. ImageNet classification with deep convolutional neural networks. In: *Proceedings of the 25th International Conference on Neural Information Processing Systems* - Volume 1, NIPS'12. Red Hook, NY, USA: Curran Associates Inc., 2012, S. 1097–1105.

[LBDH89] Lecun Y., Boser B., Denker J.S., Henderson D., Howard R.E., Hubbard W., Jackel L.D. Backpropagation applied to handwritten zip code recognition. In: *Neural Computation* Bd. 1 (1989), Nr. 4, S. 541–551.

[LDGH17] Lin T-Y., Dollár P., Girshick R., He K., Hariharan B., Belongie S. Feature Pyramid Networks for Object Detection. In: *arXiv:1612.03144* [cs] (2017).

[LFYC16] Li W., Fu H., Yu L., Cracknell A. Deep Learning Based Oil Palm Tree Detection and Counting for High-Resolution Remote Sensing Images. In: *Remote Sensing* Bd. 9 (2016), Nr. 1, S. 22.

[LKAL16] Längkvist, M., Kiselev A., Alirezaie M., Loutfi AMY. Classification and Segmentation of Satellite Orthoimagery Using Convolutional Neural Networks. In: *Remote Sensing* Bd. 8 (2016), Nr. 4, S. 329.

[LNNS16] Lukas V., Novák J., Neudert L., Svobodova I., Rodriguez-Moreno F., Edrees M., Ken J. The Combination of Uav Survey and Landsat Imagery for Monitoring of Crop Vigor in Precision Agriculture. In: *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* Bd. 41B8 (2016), S. 953–957.

[LNPK15] Lehmann J.R.K., Nieberding F., Prinz T., Knoth C. Analysis of Unmanned Aerial System-Based CIR Images in Forestry—A New Perspective to Monitor Pest Infestation Levels. In: *Forests* Bd. 6 (2015), Nr. 3, S. 594–612.

[Lobi94] Lobinger G. Die Lufttemperatur als limitierender Faktor für die Schwärmaktivität zweier rindenbrütender Fichtenborkenkäferarten, lps typographus L. undPityogenes chalcographus L. (Col., Scolytidae). In: *Anzeiger für Schädlingskunde, Pflanzenschutz, Umweltschutz* Bd. 67 (1994), Nr. 1, S. 14–17.

[LQNE20] Lobo Torres D., Queiroz Feitosa R., Nigri Happ P., Elena cué la Rosa L., Marcato Junior J., Martins J., Olã Bressan P., Gonçalves W.N. U.A.: Applying Fully Convolutional Architectures for Semantic Segmentation of a Single Tree Species in Urban Environment on High Resolution UAV Optical Imagery. In: *Sensors* Bd. 20, Multidisciplinary Digital Publishing Institute (2020), Nr. 2, S. 563.

[Ma16] Ma Z. The effects of climate stability on northern temperate forests (Ph.D). Aarhus, Denmark, 2016.

[MaGM20] Martínez-Valderrama J., Guirado E., Maestre F.T. Unraveling Misunderstandings about Desertification: The Paradoxical Case of the Tabernas-Sorbas Basin in Southeast Spain. In: *Land* Bd. 9, Multidisciplinary Digital Publishing Institute (2020), Nr. 8, S. 269.

[Meas19] Measuring Vegetation (NDVI & EVI). URL: earthobservatory.nasa.gov/features/MeasuringVegetation.

[MHVH13] Meddens A.J.H., Hicke J.A., Vierling L.A., Hudak A.T. Evaluating methods to detect bark beetle-caused tree mortality using single-date and multi-date Landsat imagery. In: *Remote Sensing of Environment* Bd. 132 (2013), S. 49–58.

[MLZY19] Ma L., Liu Y., Zhang X., Ye Y., Yin G., Johnson B.A. Deep learning in remote sensing applications: A meta-analysis and review. In: *ISPRS Journal of Photogrammetry and Remote Sensing* Bd. 152 (2019), S. 166–177.

[MMMK03] Matsugu M., Mori K., Mitari Y., Kaneda Y. Subject independent facial expression recognition with robust face detection using a convolutional neural network. In: *Neural Networks, Advances in Neural Networks Research*: IJCNN ’03. Bd. 16 (2003), Nr. 5, S. 555–559.

[Murr17] Murray C. Deep Learning CNN’s in Tensorflow with GPUs. URL: hackernoon.com/deep-learning-cnns-in-tensorflow-with-gpus-cba6efe0acc2.

[NaAV19] Natesan S., Armenakis C., Vepakomma U. RESNET-BASED TREE SPECIES CLASSIFICATION USING UAV IMAGES. In: *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*. Bd. XLII-2-W13: Copernicus GmbH, 2019, S. 475–481.

[NeHH19] Neupane B., Horanont T., Hung N.D. Deep learning based banana plant detection and counting using high-resolution red-green-blue (RGB) images collected from unmanned aerial vehicle (UAV). In: *PLoS ONE* Bd. 14 (2019), Nr. 10.

[NHLB15] Näsi R., Honkavaara E., Lyytikäinen-Saarenmaa P., Blomqvist M., Litkey P., Hakala T., Viljanen N., Kantola T., U.A.: Using UAV-Based Photogrammetry and Hyperspectral Imaging for Mapping Bark Beetle Damage at Tree-Level. In: *Remote Sensing* Bd. 7 (2015), Nr. 11, S. 15467–15493.

[OJRJ21] Osco L.P., Junior J.M., Ramos A.P.M., Jorge L.A. de C., Fatholahi S.N., Silva J. de A., Matsubara E.T., Pistori H., U.A.: A Review on Deep Learning in UAV Remote Sensing. In: *arXiv:2101.10861* [cs] (2021).

[OnIs18] Onishi M., Ise T. Automatic classification of trees using a UAV onboard camera and deep learning. In: *arXiv:1804.10390* [cs, stat] (2018).

[OnIs21] Omishi M., Ise T. Explainable identification and mapping of trees using UAV RGB image and deep learning. In: *Scientific Reports* Bd. 11, Nature Publishing Group (2021), Nr. 1, S. 903.

[OrBK13] Ortoz S.M., Breidenbach J., Kändler G. Early Detection of Bark Beetle Green Attack Using TerraSAR-X and RapidEye Data. In: *Remote Sensing* Bd. 5 (2013), Nr. 4, S. 1912–1931.

[PaKi12] Paine D.P., Kiser J.D. Aerial photography and image interpretation. Chichester, UK: Wiley, 2012 — ISBN 978-1-118-11099-7.

[PiSc20] Pisner D.A., Schnyer D.M. Chapter 6 - Support vector machine. In: MECHELLI, A.; VIEIRA, S. (Hrsg.): Machine Learning: Academic Press, 2020 — ISBN 978-0-12-815739-8, S. 101–121.

[PKUB18] Pashenova N.V., Kononov A.V., Ustyantsev K.V., Blinov A.G., Pertsovaya A.A., Baranchikov Y.N. Ophiostomatoid Fungi Associated with the Four-Eyed Fir Bark Beetle on the Territory of Russia. In: Russian Journal of Biological Invasions Bd. 9 (2018), Nr. 1, S. 63–74.

[QGWC20] Qiao R., Ghodsi A., Wu H., Chang Y., Wang C. Simple weakly supervised deep learning pipeline for detecting individual red-attacked trees in VHR remote sensing images. In: *Remote Sensing Letter*s Bd. 11, Taylor & Francis (2020), Nr. 7, S. 650–658.

[QiGr21] Qin R., Gruen A. The role of machine intelligence in photogrammetric 3D modeling – an overview and perspectives. In: *International Journal of Digital Earth* Bd. 14, Taylor & Francis (2021), Nr. 1, S. 15–31.

[RaYa17] Razavi S., Yalcin H. Using convolutional neural networks for plant classification. In: 2017 25th Signal Processing and Communications Applications Conference (SIU), 2017, S. 1–4.

[RDGF16] Redmun J., Divvala S., Girshick R., Farhadi A. You Only Look Once: Unified, Real-Time Object Detection. In: *arXiv:1506.02640* [cs] Bd. 56 (2016).

[ReFa16] Redmon J., Farhadi A. YOLO9000: Better, Faster, Stronger. In: *arXiv:1612.08242* [cs] (2016).

[ReFa18] Redmon J., Farhadi A. YOLOv3: An Incremental Improvement. In: *arXiv:1804.02767* [cs] (2018).

[Reic06] Reichhardt T. The First Photo From Space. URL: airspacemag.com/space/the-first-photo-from-space-13721411/.

[RHGS16] Ren S., He K., Girshick R., Sun J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In: *arXiv:1506.01497* [cs] (2016).

[RTGS19] Rezatofighi H., Tsoi N., Gwak J., Sadeghian A., Reid I., Savarese S. Generalized Intersection over Union: A Metric and A Loss for Bounding Box Regression. In: *arXiv:1902.09630* [cs] (2019).

[Sasa07] Sasaki Y. The truth of the F-measure. In: Teach Tutor Mater Bd. 1 (2007), Nr. 5, S. 1–5.

[Scag19] Scagliarini M. Xylella, l'Ue cambierà le misure di emergenze: ridotta l'area di taglio. URL: lagazzettadelmezzogiorno.it/news/home/1184219/xylella-l-ue-cambiera-le-misure-di-emergenze-ridotta-l-area-di-taglio.html.

[Scho06] Schowengerdt R. Remote Sensing. Models and Methods for Image Processing. 3rd Edition. University of Arizona, Dept. of Electrical and Computer Engineering: *Academic Press*, 2006 — ISBN 978-0-12-369407-2.

[SGMA21] Safonova A., Guirado E., Maglinets Y., Alcaraz-Segura D., Tabik S. Olive Tree Biovolume from UAV Multi-Resolution Image Segmentation with Mask R-CNN. In: *Sensors* Bd. 21, Multidisciplinary Digital Publishing Institute (2021), Nr. 5, S. 1617.

[SIVA16] Szegedy C., Ioffe S., Vanhoucke V., Alemi A. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. In: *arXiv:1602.07261* [cs] (2016).

[SMAD18] Samanta S., Mukherjee A., Ashour A.S., Dey N., Tavares J.M., Abdessalem Karâa W.B., Taiar R., Azar A.T., U.A.: Log Transform Based Optimal Image Enhancement Using Firefly Algorithm for Autonomous Mini Unmanned Aerial Vehicle: An Application of Aerial Photography. In: *International Journal of Image and Graphics* Bd. 18, World Scientific Publishing Co. (2018), Nr. 04, S. 1850019.

[SMAD19] Dos Santos A.A., Marcato Junior J., Araújo M.S., Di Martin D.R., Tetila E.C., Siqueira H.L., Aoki C., Eltner A., U.A.: Assessment of CNN-Based Methods for Individual Tree Detection on Images Captured by RGB Cameras Attached to UAVs. In: *Sensors* (Basel, Switzerland) Bd. 19 (2019), Nr. 16.

[SMFD08] Sofo A., Manfreda S., Fiorentino M., Dichio B., Xiloyannis C. The olive tree: a paradigm for drought tolerance in Mediterranean climates. In: *Hydrology and Earth System Sciences* Bd. 12 (2008), Nr. 1, S. 293–301.

[STAR19] Safonova A., Tabik S., Alcaraz-Segura D., Rubtsov A., Maglinets Y., Herrera F. Detection of Fir Trees (*Abies sibirica*) Damaged by the Bark Beetle in Unmanned Aerial Vehicle Images with Deep Learning. In: *Remote Sensing* Bd. 11 (2019), Nr. 6, S. 643.

[StKa20] Stateras D., Kalivas D. Assessment of Olive Tree Canopy Characteristics and Yield Forecast Model Using High Resolution UAV Imagery. In: *Agriculture* Bd. 10, Multidisciplinary Digital Publishing Institute (2020), Nr. 9, S. 385.

[Sunc20] SunCalc sun position and sun phases calculator. URL: suncalc.org.

[SyDB19] Sylvain J-D., Drolet G., Brown N. Mapping dead forest cover using a deep convolutional neural network and digital aerial photography. In: *ISPRS Journal of Photogrammetry and Remote Sensing* Bd. 156 (2019), S. 14–26.

[TAFG19] Tsagkatakis G., Aidini A., Fotiadou K., Giannopoulos M., Pentari A., Tsakalides P. Survey of Deep-Learning Approaches for Remote Sensing Observation Enhancement. In: *Sensors* Bd. 19, Multidisciplinary Digital Publishing Institute (2019), Nr. 18, S. 3929.

[Tens20] TensorFlow 2 Object Detection API tutorial — TensorFlow 2 Object Detection API tutorial documentation. URL: tensorflow-object-detection-api-tutorial.readthedocs.io/en/latest/.

[TLZD20] Tao H., Li C., Zhao D., Deng S., Hu H., Xu X., Jimg W. Deep learning-based dead pine tree detection from unmanned aerial vehicle images. In: *International Journal of Remote Sensing* Bd. 41, Taylor & Francis (2020), Nr. 21, S. 8238–8255.

[TPHH17] Tabik S., Peralta D., Herrera-Poyatos A., Herrera F. A snapshot of image pre-processing for convolutional neural networks: case study of MNIST. In: *International Journal of Computational Intelligence Systems* Bd. 10, Atlantis Press (2017), Nr. 1, S. 555–568.

[TrDL20] Trang N.H., Diez Y., Lopez L. Insect Damaged Tree Detection with Drone Data and Deep Learning Technique, Case Study: Abies Mariesii Forest, Zao Mountain, Japan Bd. 22 (2020), S. 17917.

[UBSA20] Ulku I., Barmpoutis P., Stathaki T., Akagunduz E. Comparison of single channel indices for U-Net based segmentation of vegetation in satellite images. In: *Twelfth International Conference on Machine Vision (ICMV 2019)*. Bd. 11433: International Society for Optics and Photonics, 2020, S. 1143319.

[USGS13] Uijlings J.R.R., Van de Sande K.E.A., Gevers T., Smeulders A.W.M. Selective Search for Object Recognition. In: *International Journal of Computer Vision* Bd. 104 (2013), Nr. 2, S. 154–171.

[VaBC20] Varkarakis V., Bazrafkan S., Corcoran P. Deep neural network and data augmentation methodology for off-axis iris segmentation in wearable headsets. In: *Neural Networks* Bd. 121 (2020), S. 101–121.

[WaBL21] Wang C-Y., Bochkovskiy A., Laio H-Y.M. Scaled-YOLOv4: Scaling Cross Stage Partial Network. In: *arXiv:2011.08036* [cs] (2021).

[WBPS12] Wolter P.T., Berkley E.A., Peckham S.D., Singh A., Townsend P.A. Exploiting tree shadows on snow for estimating forest basal area using Landsat data. In: *Remote Sensing of Environment* Bd. 121 (2012), S. 69–79.

[Werm04] Wermelinger B. Ecology and management of the spruce bark beetle Ips typographus—a review of recent research. In: *Forest Ecology and Management* Bd. 202 (2004), Nr. 1, S. 67–82.

[WFSH18] Wagner F.H., Ferreira M.P., Sanchez A., Hirye M.C.M., Zortea M., Gloor E., Phillips O.L., de Souza Filho C.R., U.A.: Individual tree crown delineation in a highly diverse tropical forest using very high resolution satellite images. In: *ISPRS Journal of Photogrammetry and Remote Sensing* Bd. 145 (2018), S. 362–377.

[WKJS14] Waser L.T., Küchler M., Jütte K., Stampfer T. Evaluating the Potential of WorldView-2 Data to Classify Tree Species and Different Levels of Ash Mortality. In: *Remote Sensing* Bd. 6 (2014), Nr. 5, S. 4515–4545.

[WWSD19] Wu H., Wiesner-Hanks T., Stewart E.L., Dechant C., Kaczmar N., Gore M.A., Nelson R.J., Lipson H. Autonomous Detection of Plant Disease Symptoms Directly from Aerial Imagery. In: *The Plant Phenome Journal* Bd. 2, American Society of Agronomy and Crop Science Society of America (2019), Nr. 1.

[XATR20] Xia Y., D'angelo P., Tian J., Reinartz P. Dense matching comparison between classical and deep learning based algorithms for remote sensing data. In: *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences.* Bd. XLIII-B2-2020: Copernicus GmbH, 2020, S. 521–525.

[YJCA19] Yeom J., Jung J., Chang A., Ashapure A., Maeda M., Maeda A., Landivar J. Comparison of Vegetation Indices Derived from UAV Data for Differentiation of Tillage Effects in Agriculture. In: *Remote Sensing* Bd. 11, Multidisciplinary Digital Publishing Institute (2019), Nr. 13, S. 1548.

[ZhTZ19] Zhang W., Tang P., Zhao L. Remote Sensing Image Scene Classification Using CNN-CapsNet. In: *Remote Sensing* Bd. 11 (2019), Nr. 5, S. 494.

[ZQHF15] Zhang Q., Qin R., Huang X., Fang Y., Liu L. Classification of Ultra-High Resolution Orthophotos Combined with DSM Using a Dual Morphological Top Hat Profile. In: *Remote Sensing* Bd. 7, Multidisciplinary Digital Publishing Institute (2015), Nr. 12, S. 16422–16440.

[ZTMX17] Zhu X.X., Tiua D., Mou L., Xia G., Zhang L., Xu F., Fraundorfer F. Deep Learning in Remote Sensing: A Comprehensive Review and List of Resources. In: *IEEE Geoscience and Remote Sensing Magazine* Bd. 5 (2017), Nr. 4, S. 8–36.

[ZYNW18] Zhao T., Yang Y., Niu H., Wang D., Chen Y. Comparing U-Net convolutional network with mask R-CNN in the performances of pomegranate tree canopy segmentation. In: *Multispectral, Hyperspectral, and Ultraspectral Remote Sensing Technology, Techniques and Applications VII*. Bd. 10780. Honolulu, Hawaii, United States: International Society for Optics and Photonics, 2018, S. 107801J.

[Лер97] Лер П.А. Определитель насекомых Дальнего Востока России. Т. 05. Ручейники и чешуекрылые. Часть 01 [DJVU], 1. Bd. 05. *Владивосток: Дальнаука*, 1997.

[Рябо13] Рябовол С.В. Растительность г. Красноярска - Современные проблемы науки и образования (научный журнал) (2013). — Электронный научный журнал. *Современные проблемы науки и образования*.