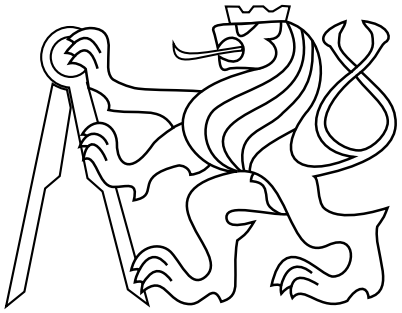
Czech Technical University in Prague

Faculty of Electrical Engineering
Department of Cybernetics
Multi-robot Systems



Detection of Bark-Beetle Infestation Using an Autonomous UAV

Master's Thesis

Tereza Uhrová

Prague, May 2023

Study programme: Cybernetics and Robotics
Supervisor: Ing. Matouš Vrba

Acknowledgments

I would like to express my sincere gratitude to Ing. Matouš Vrba, my supervisor, for his guidance, support, and invaluable feedback throughout the course of this project. I would also like to thank Professor Jiří Matas for his kind assistance and expert advice.

I am grateful to my father for his help in acquiring and transporting the tree log samples, without which this research would not have been possible.

Lastly, I would like to extend my heartfelt thanks to my boyfriend, friends and family for their unwavering support, encouragement, and understanding throughout my academic journey.

Author Statement

I declare that the presented work was developed independently and that I have listed all sources of information used within it in accordance with the methodical instructions for observing the ethical principles in the preparation of university theses.

Prague, May 26. 2023 .....

Tereza Uhrová

MASTER‘S THESIS ASSIGNMENT

I. Personal and study details

Student's name: **Uhrová Tereza** Personal ID number: **483546**

Faculty / Institute: **Faculty of Electrical Engineering**

Department / Institute: **Department of Cybernetics**

Study program: **Cybernetics and Robotics**

II. Master’s thesis details

Master’s thesis title in English:

Detection of Bark-Beetle Infestation Using an Autonomous UAV

Master’s thesis title in Czech:

Detekce stromů nakažených kůrovcem pomocí autonomního bezpilotního letounu

Guidelines:

The aim of this thesis is to develop a system for detection of trees infested with the European spruce bark beetle using an autonomous UAV flying below the tree canopy and equipped with an RGB camera and a LiDAR.
Research different methods of machine perception with a focus on visual detection of small objects. Design and implement a system for tree diagnosis. The input of the system will be data from the UAV's onboard sensors and the output should be a diagnosis of the individual trees observed by the UAV together with a corresponding confidence measure. Gather data in real-world environments for training the system and testing it. Evaluate its performance on the gathered data and in realistic simulations.

Bibliography / sources:

[1] O. Ronneberger, P. Fischer, T. Brox, „U-net: Convolutional networks for biomedical image segmentation,“International Conference on Medical image computing and computer-assisted intervention, 2015.
[2] X. Zhou, D. Wang and P. Krähenbühl, „Objects as Points“, arXiv 1904.07850, 2019.
[3] T. Klouček, J. Komárek, P. Surový, K. Hrach, P. Janata, and B. Vašíček, „The use of UAV mounted sensors for precise detection of bark beetle infestation,“ Remote Sensing, 11(13), p.1561, 2019.

Name and workplace of master’s thesis supervisor:

Ing. Matouš Vrba Multi-robot Systems FEE

Name and workplace of second master’s thesis supervisor or consultant:

Date of master’s thesis assignment: **10.02.2023** Deadline for master's thesis submission: **26.05.2023**

Assignment valid until: **22.09.2024**

Ing. Matouš Vrba	prof. Ing. Tomáš Svoboda, Ph.D.	prof. Mgr. Petr Páta, Ph.D.
Supervisor’s signature	Head of department’s signature	Dean’s signature

III. Assignment receipt

The student acknowledges that the master’s thesis is an individual work. The student must produce her thesis without the assistance of others, with the exception of provided consultations. Within the master’s thesis, the author must state the names of consultants and include a list of references.

. _____
Date of assignment receipt

_____
Student’s signature

Abstract

This thesis presents a UAV-based system for the detection of European spruce bark beetle-infested trees. The system is designed to identify small holes made by the bark beetle as it drills into the tree's bark and phloem, allowing for early detection of infestations. The pipeline consists of three stages: tree trunk segmentation, hole detection, and final classification. Integration of a high-resolution RGB camera and a LiDAR sensor enables detailed image capture and accurate distance measurements.

The tree trunk segmentation stage employs a ResNet50 network trained to segment pixels corresponding to tree trunks in RGB images. For the hole detection, a Maximally Stable Extremal Regions (MSER) blob detection algorithm is applied, enhanced by histogram equalization and filtering based on circularity and intensity properties of the detected blobs. Additionally, a YOLOv7 model is trained to compare it with the proposed detector. The final classification utilizes the "holes per area" metric, the ratio of the number of detected holes to the visible bark area. A tree is classified using this metric based on histograms of healthy and infected trees in the training dataset.

The developed system demonstrates its effectiveness in early detection and monitoring of European spruce bark beetle infestations, providing forest managers with a valuable tool for proactive forest health management and minimizing economic losses associated with bark beetle outbreaks.

Keywords Unmanned Aerial Vehicles, Machine Vision, Bark beetle infestation, Machine learning

Abstrakt

Tato práce prezentuje UAV systém, který detekuje stromy nakažené lýkožroutem smrkovým. Tento systém je navržen, aby identifikoval malé dírky, které vytvořil lýkožrout, když se zavrtal do kůry. Díky tomu jsou nakažené stromy detekovány velmi brzo po nákaze. Úloha je rozdělena na tři části: segmentace kmene stromu, detekce děr a finální klasifikace. Použití RGB kamery s vysokým rozlišením a LiDAR senzoru umožňuje zachycení detailních obrázků a přesné měření vzdálenosti.

Segmentace kmene stromu používá sítě natrénované ResNet50, aby segmentovala pixely odpovídající kmeni v RGB obrázcích. Co se týče detekce děr, je použit algoristmus MSER (Maximally Stable Extremal Regions), který detekuje podezřelé skvrny. Spolu s ekvalizací histogramu a filtrováním založeném na kulatosti a intenzitě jsme schopni detekovat dírky způsobené lýkožroutem. Dodatečně, je natrénována síť YOLOv7 a porovnána s navrženým detektorem. Finální klasifikace používá "díry na plochu" metriku, poměr počtu detekovaných děr k viditelné ploše kůry. Strom je klasifikován pomocí této metriky a histogramů nakažených a zdravých stromů z trénovacích dat.

Vyvinutý systém demonstruje svou efektivitu v brzké detekci a kontrole lýkožrouta smrkového a poskytuje lesním hospodářům cenný nástroj v boji proti lýkožroutu.

Klíčová slova Bezpilotní Prostředky, Strojové vidění, Kůrovcová kalamita, Strojové učení

Abbreviations

CLAHE Contrast Limited Adaptive Histogram Equalization

CNN Convolutional Neural Network

COCO Common Objects in Context

DARPA Defense Advanced Research Projects Agency

DBH Diameter at Breast Height

FASTER Fast and Safe Trajectory Planner

FOV Field of View

gLoG generalized Laplacian of Gaussian

GPS Global Positioning System

IMU Inertial Measurement Unit

IoU Intersection over Union

LiDAR Light Detection and Ranging

mIoU mean Intersection over Union

MLS Mobile Laser Scanning

MRI Magnetic Resonance Imaging

MRS Multi-robot Systems Group

MSER Maximally Stable Extremal Regions

R-CNN Residual Convolutional Neural Network

SLAM Simultaneous Localization And Mapping

UAV Unmanned Aerial Vehicle

YOLO You Only Look Once

Contents

Chapter 1

Introduction

1.1 Motivation

Bark beetle infestations pose a serious threat to forests in the Czech Republic (see Figure 1.1), causing irreparable damage to trees and ecosystems [8]. Currently, traditional ground-based methods, such as manual visual inspection with telescopes, are used to monitor for infestations, but they can be time-consuming and labor-intensive [18]. Early detection of infestations is crucial, as it allows for control measures to be taken before new generations of beetles fully develop in the tree phloem.

Figure 1.1: A dead forest in Šumava mountains, caused by bark beetle[1].

In the Czech Republic, one of the most destructive bark beetle species is Ips typographus (see Figure 1.2), which feeds on the phloem of Norway spruces, the dominant coniferous tree species. Infestations of this beetle can cause significant damage, including tree mortality and increased risk of wildfire. In recent years, the use of autonomous UAVs has emerged as a promising solution for early detection of bark beetle infestations, offering a more efficient and cost-effective means of monitoring forest health.

Despite the potential advantages of UAVs, few studies have investigated their use in detecting bark beetle infestations (see section 1.2.1), particularly in areas with dense tree canopies. While existing UAV systems use hyperspectral cameras to identify unhealthy trees from above the canopies, there are limitations to localization of those infected trees when they are supposed to be taken down by disposal team. In this thesis, we propose a novel system for detecting Ips typographus infestations using an autonomous UAV that flies under the tree canopies and scans the tree trunks.

Our system is designed to identify small holes made by the bark beetle as it drills into the tree's bark and phloem, allowing for early detection of infestations before significant damage

[1]Source: https://www.vulhm.cz/files/uploads/2019/02/TZ_Sumava_vyzkum_r1.pdf

Figure 1.2: European bark beetle (Ips typographus)[2].

occurs. This represents a significant improvement over traditional ground-based monitoring methods, which can be hindered by obstacles such as dense vegetation or steep terrain. By providing a more comprehensive and accurate view of the forest canopies, our proposed system has the potential to revolutionize the way in which bark beetle infestations are detected and managed.

1.2 Related Work

To our best knowledge there are no published papers that would take the same approach to tree diagnosis by UAVs under the canopy as us. The task of this thesis is divided into smaller problems which are represented as thematic subsections.

1.2.1 UAV Tree Diagnosis Systems

Numerous studies have been carried out on the diagnosis of tree diseases, indicating that the most efficient method of covering a large area is through the use of aerial imagery. Hyperspectral cameras have been identified as an excellent tool for this purpose.

The research presented in [17] investigated the potential of using hyperspectral data to detect early stages of bark beetle infestation in Norway spruce trees. They found significant differences in the spectral response and foliar biochemical properties (chlorophyll and nitrogen concentration) between healthy and infested trees. The results suggest that hyperspectral remote sensing can be used for early detection of bark beetle infestation. These properties are utilized by the subsequent research.

Several studies have investigated the use of UAV systems for tree diagnosis, with a focus on detecting signs of tree stress and disease. In [11], an autonomous UAV equipped with a hyperspectral camera was used to scan the canopy of a forest. The authors applied Convolutional Neural Network (CNN) to detect different tree species and conditions, including healthy spruces, early-stage infected spruces, and dead spruces. The system achieved an F-1 score of 90 % in detecting infected trees.

[2]Source: `https://upload.wikimedia.org/wikipedia/commons/1/12/Ips.typographus.jpg`

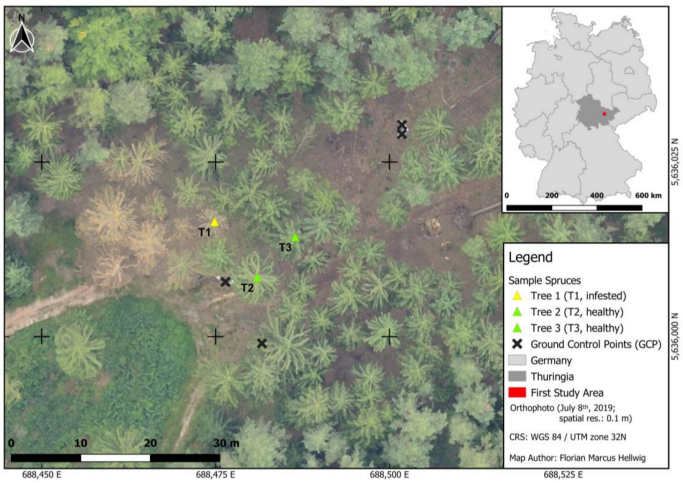


Figure 1.3: Example of detection of individual trees by a hyperspectral camera [7].

Another study [7] investigated the use of airborne hyperspectral data and field spectrometer measurements to detect early spruce infestation by bark beetles in Germany's spruce forests (see Figure 1.3). The researchers defined a new hyperspectral index and compared it with other common indices for bark beetle detection. The new index showed a very high Overall Accuracy (OAA) of 98.84 % and the ability to detect a larger proportion of infested spruces in the early infestation phase compared to commonly used indices.

[6] explores the potential of Sentinel-2 satellite data to detect early stages of bark beetle infestation in Norway spruce monoculture forests in the Czech Republic. The authors constructed seasonal trajectories of vegetation indices from multi-temporal observations and used a random forest algorithm to classify healthy and infested trees.

There are two main challenges using these methods. First, precisely locating trees suspected to be infected is difficult. Second, following consultations with specialists from Lesy ČR, it was revealed that methods utilizing vegetation indices may only detect sick trees and not necessarily those freshly infected by bark beetles, making them unreliable for our purposes

1.2.2 Tree Trunk Segmentation

The process of accurately identifying and extracting the tree trunk region from an RGB image or Light Detection and Ranging (LiDAR) data is a challenging task. Over the years, numerous studies have been conducted to tackle this problem using a variety of approaches and sensors.

The authors of [9] propose a method for real-time segmentation of tree trunks in municipal regions using a semantic segmentation CNN architecture that incorporates a depthwise residual block to reduce the number of network parameters. The method also performs post-processing to refine the segmented regions and extract the central line of the identified region for future computation of trunk measurements (see Figure 1.4). The proposed approach is robust and achieves similar evaluation metrics as the original U-Net architecture [27], but with significant reduction in network size.

The second study focuses on the automated training of fruit trees to adapt to fruiting walls in modern orchard architectures using trunk and branch segmentation. In [19], a

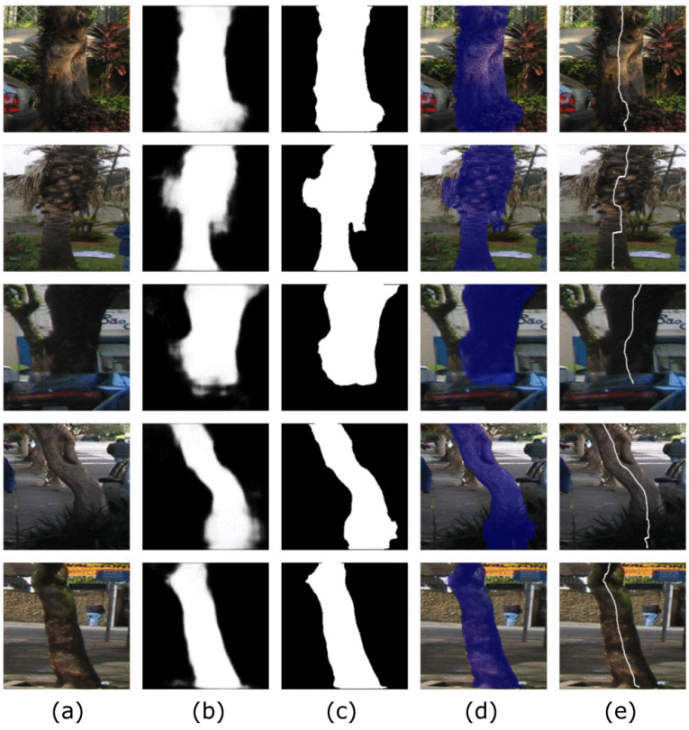


Figure 1.4: Results of tree trunk segmentation [9].

trunk and branch segmentation method using a Kinect V2 sensor and deep learning-based semantic segmentation is proposed. Point cloud data from the sensor are used to remove background trees, and a CNN (SegNet) is used for semantic segmentation of the tree's trunk and branches. The study achieves high segmentation accuracy and mean Intersection over Union (mIoU) scores, demonstrating the potential of deep learning-based semantic segmentation for automated branch detection in orchard environments.

Liu et al. propose a method for improving the accuracy of single tree segmentation and Diameter at Breast Height (DBH) estimation using Mobile Laser Scanning (MLS) in complex forest environments in [10]. The proposed method includes relative density segmentation for tree trunk segmentation and a multi-height diameter-based DBH estimation method with outlier detection. The results show high precision and recall rates for tree segmentation and low mean absolute error and root mean square error for DBH estimation.

It is worth noting that in our work, we consider RGB images from a stabilized camera, which is difficult to synchronize with LiDAR data. Therefore, we choose to train a CNN to segment a tree trunk in the foreground, keeping in mind that the background consists of numerous other trees and greenery. These studies show the potential for using deep learning-based segmentation methods to automate tree-related tasks in various environments, such as urban areas, orchards, and forests.

1.2.3 Small Object Detection

Detecting small holes in tree bark is a challenging problem due to their size and the complexity of the bark's texture. In recent years, there have been several studies on detecting small structure of similar color, commonly called "blobs", in various types of images, including

medical images, images of nature and images from industrial inspection. A blob is defined as a convex area with similar pixel intensity in the image.

Kong et al. present a generalized Laplacian of Gaussian (gLoG) filter for detecting elliptical blob structures in images in [28]. The gLoG filter can locate blob centers and estimate scales, shapes, and orientations by generalizing the common 3-D LoG scale-space blob detector to a 5-D gLoG scale-space one. The filter finds local maxima of an intermediate map obtained by aggregating the log-scale-normalized convolution responses of each individual gLoG filter, instead of searching for local extrema of the image's 5-D gLoG scale space. The filter was tested on both biomedical and natural images, accurately detecting cell nuclei and estimating texture orientations.

In [24], it is proposed gLoG for detecting small structures in medical images, particularly for detecting kidney glomeruli in 3D Magnetic Resonance Imaging (MRI) scans. gLoG is designed to efficiently segment blobs based on local convexity, intensity, and shape information. The algorithm first pre-segments the image into blob candidate regions and then extracts three novel features (regional blobness, regional flatness, and regional intensity) from these regions. These features are used in an unsupervised learning algorithm for post-pruning. The paper demonstrates the effectiveness of gLoG in detecting large numbers of small blobs through simulated images and real kidney 3D MRI images, verifying its robustness and efficiency for 3D blob segmentation.

The authors of [12] propose a joint constraint blob detector using U-Net and Hessian analysis to accurately detect small objects in medical images. The proposed approach, UH-DoG, is evaluated on fluorescent and kidney MRI datasets and outperforms other methods in precision and F-score. The paper highlights the importance of accurate blob detection in developing imaging biomarkers for disease diagnosis and staging.

A Residual Convolutional Neural Network (R-CNN) is used in [22] to detect small objects. Specifically, authors examine influence of object's size on results. Then, a modified Faster R-CNN is proposed which is able to improve the overall detection performance on a real world dataset for company logo detection.

YOLOv7, proposed in [4], is an efficient object detector that surpasses other real-time detectors in terms of both speed and accuracy. It outperforms transformer-based and other convolutional-based detectors by a significant margin.

In this section, we reviewed several studies related to small blob detection, including the gLoG filter, gLoG, joint constraint blob detector using U-Net and Hessian analysis, and modified Faster R-CNN models. Additionally, the development of efficient object detectors like YOLOv7 has significantly improved both speed and accuracy in object detection tasks. Although these studies mainly focus on detecting small structures in medical images, they have a potential for detecting small holes in the tree bark. However, due to the complexity of the bark's texture and the variation in lighting conditions, further research is needed to develop a robust and accurate method for detecting bark beetle-infected trees from RGB images.

1.2.4 UAV Mapping and Localization in Forests

In recent years, the development of autonomous aerial robots has been a focus of research in robotics. An important aspect of autonomous aerial robots is their ability to navigate through unknown environments with high speed and accuracy. In this section, we review three papers that propose different approaches to achieve this goal.

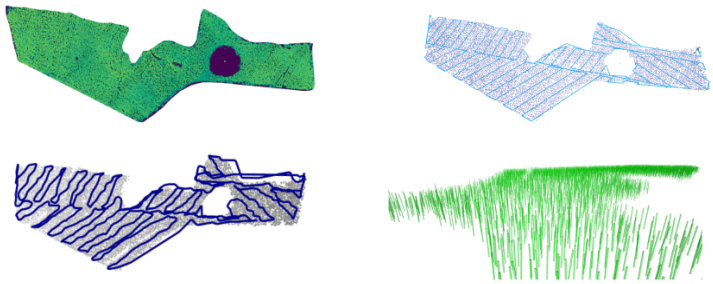


Figure 1.5: Example of a plan to map 0.77 km^2 of forest by [3].

An integrated system that can perform large-scale autonomous flights and real-time semantic mapping in challenging forest environments under the tree canopies was proposed in [3]. The authors detect and model tree trunks and ground planes from LiDAR data and use them to constrain the collision-free robot poses during planning. The autonomous navigation module utilizes a multi-level planning and mapping framework to compute dynamically feasible trajectories that lead the UAV to build a semantic map of the user-defined region of interest (see Figure 1.5). The proposed drift-compensation mechanism minimizes the odometry drift using semantic Simultaneous Localization And Mapping (SLAM) outputs in real-time while maintaining planner optimality and controller stability. The proposed system can execute its mission accurately and safely at scale.

The Fast and Safe Trajectory Planner (FASTER) planning algorithm is proposed in [16]. FASTER enables the local planner to optimize in both the free-known and unknown spaces, which results in high-speed trajectories. Safety is ensured by always having a feasible, safe back-up trajectory in the free-known space at the start of each replanning step. The proposed algorithm is tested extensively both in simulation and in real hardware, showing agile flights in unknown cluttered environments with velocities up to 3.6 $\mathrm{m\,s}^{-1}$.

In [20], a quadrotor system capable of fast autonomous flight is presented. The system can navigate through mixed indoor and outdoor environments at speeds of more than 18 $\mathrm{m\,s}^{-1}$. The authors provide an overview of their system and details about the specific component technologies that enable the high-speed navigation capability of their platform. The system's robustness is demonstrated through high-speed autonomous flights and navigation through a variety of unknown obstacle-rich environments.

This research in [1] presents a novel approach for autonomous UAVs in search and rescue operations in complex subterranean environments. The proposed system ranked second in the Defense Advanced Research Projects Agency (DARPA) SubT Finals and proved to be robust for real-world deployment. The approach enables fully autonomous and decentralized UAV team deployment with seamless simulation-to-world transfer. The main contributions lie in the mapping and navigation pipelines, utilizing novel map representations and methods for efficient planning and exploration. Extensive experimental verification supports the performance of the UAV system in diverse environments.

As we have determined that the issue of high-speed flight in unknown cluttered environments is outside the scope of our thesis and has already been addressed by existing research, it is not the focus of this thesis.

1.3 Problem Specification

The aim of this thesis is to develop a system that can detect European spruce bark beetle (Ips typographus) infestations in trees using an UAV equipped with an RGB camera and a LiDAR sensor. We assume that obstacle mapping, Global Positioning System (GPS) denied self-localization and collision-free trajectory planning for UAVs in forests are already solved and that locations of suspected infestation are identified either by a different system or by a human expert.

The system is designed to fly autonomously under the tree canopy, while the sensors scan each tree trunk to gather data. The system then analyzes this data and provides a diagnosis of each individual tree it observes. Specifically, it outputs a confidence measure indicating whether or not the tree is infected with the bark beetle. This is accomplished by detecting small holes made by the beetle as it drills into the tree's bark and phloem, allowing for early detection of infestations before significant damage occurs.

Chapter 2

Pipeline overview

The developed system consists of three main components as seen in Figure 2.1. The first component processes raw images captured by the UAV's RGB camera, identifying tree trunks and extracting patches of the trunk. The second component detects the small entry holes made by the European spruce bark beetle in the tree's bark. The final part evaluates detected holes and outputs confidence measure if the tree is infected or not.

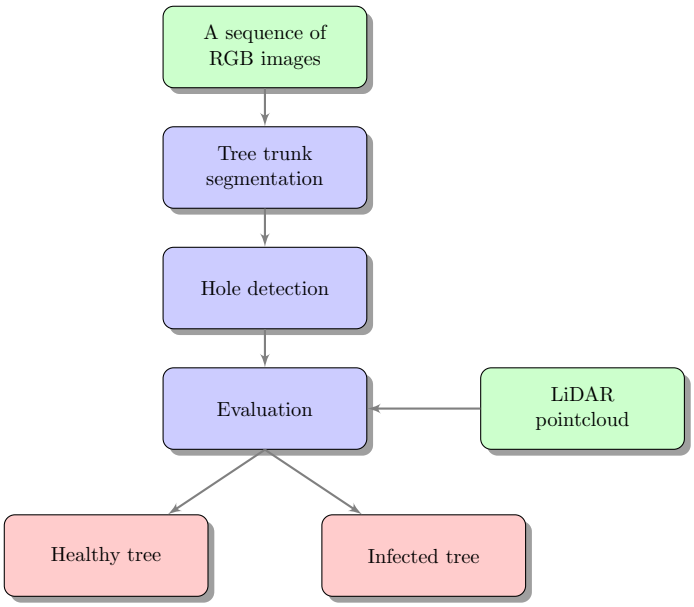


Figure 2.1: Flowchart of the system's pipeline. Green boxes are inputs and red boxes indicate outputs of the system.

The first part of the system uses a CNN to segment the trunk of a single tree in the foreground. It is described in Chapter 3. The second part of the system employs the MSER [29] algorithm to detect blobs that are subsequently filtered in Chapter 4. The last step, presented in Chapter 5, computes a confidence measure from "holes per area" if the tree is infected or not.

2.1 Hardware

A custom-built quadrotor UAV is utilized for data collection and analysis (see Figure 2.2). This UAV relies on the Multi-robot Systems Group (MRS) UAV platform [2]. It

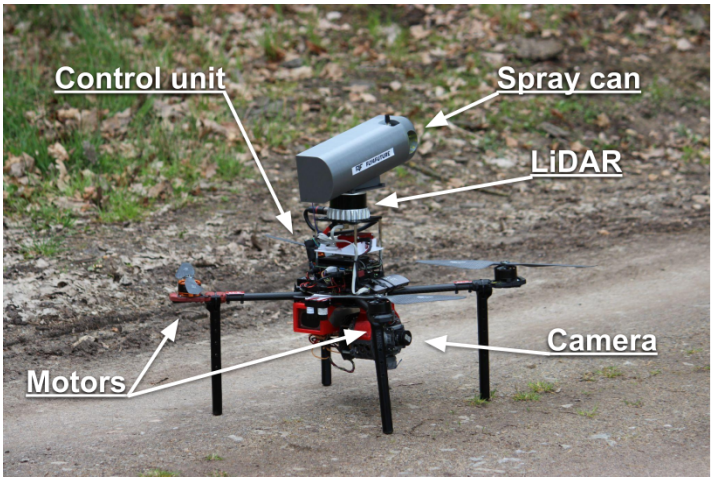


Figure 2.2: The UAV equipped for bark beetle detection used for the experiments.

is equipped with a variety of hardware components to ensure efficient and accurate data collection.

To stabilize the camera during flight, a custom-made gimbal is installed on the UAV. This gimbal is specifically designed to hold the FLIR Blackfly S GigE camera, which has a horizontal Field of View (FOV) of 16 degrees and a vertical FOV of 19 degrees. The camera is used to capture high-resolution images of the trees, which are later analyzed to identify trees infected with bark beetle.

In addition to the camera, a custom-made system is integrated into the UAV to spray paint on trees infected with the bark beetle. This system is designed to mark trees for further investigation and treatment.

To aid in mapping and navigation, a LiDAR sensor and an Inertial Measurement Unit (IMU) are installed on the UAV. These sensors are critical for self-localization, stabilization and collision-free flight which is provided by the MRS UAV system [5].

Overall, the UAV used in this experiment is a highly specialized piece of hardware, designed specifically for the task of collecting, analyzing data and marking infected trees with bark-beetle. However, the design of the UAV is not part of this thesis.

Chapter 3

Tree Trunk Segmentation

As the first operation, tree trunk of the tree being inspected is located in the image. The background typically consists of more trees and vegetation, so to separate it, we train a CNN to segment tree trunks in foreground and the background. To do this, a custom dataset was captured and manually annotated.

3.1 Dataset

To create the tree trunk segmentation dataset, we began by utilizing online images, but later enhanced the dataset with our own photographs captured within the forests near Vimperk, Sušice, and Temešvár. The resulting dataset consists of 343 annotated images (see Figure 3.1). The annotation of data for segmentation was performed in the Common Objects in Context (COCO) format [26] using the online tool Datatorch[1]. The dataset was randomly divided into training, validation and testing subsets. Training subset is 80 % of images, validation and testing 10 % each.

3.2 Training

Training a convolutional neural network for segmenting tree trunks and background requires careful selection of network architecture and training parameters. In our study, PyTorch [15] library, which provides a range of tools for building, training, and evaluating deep neural networks, is used. Our network is trained on a remote GPU provided by Google Colab, which allowed us to take advantage of high-performance computing resources without the need for local hardware.

To ensure that our network was able to accurately segment tree trunks and background, pretrained segmentation networks provided by PyTorch are used. These networks are trained on large-scale datasets and can be fine-tuned for specific tasks using transfer learning. We selected the DeepLabV3 [21] model as our base architecture due to its superior performance in image segmentation tasks.

To further enhance the performance of our network, we experimented with different backbones for the DeepLabV3 model. Specifically, we tested the ResNet50 [25], ResNet101 [25], and MobileNetV3-Large [13] backbones. These backbones differ in their depth, number of layers, and computational complexity, which can affect the accuracy and speed of the network. By comparing the performance of these three options, we aimed to identify the optimal backbone for our custom dataset.

[1] https://datatorch.io

(a) Example picture

(b) Example picture with annotated mask

(c) Example picture

(d) Example picture with annotated mask

Figure 3.1: Example pictures from the tree trunk segmentation dataset. Fig. 3.1a and Fig. 3.1c are the original images, the Fig. 3.1b and Fig. 3.1d are annotated.

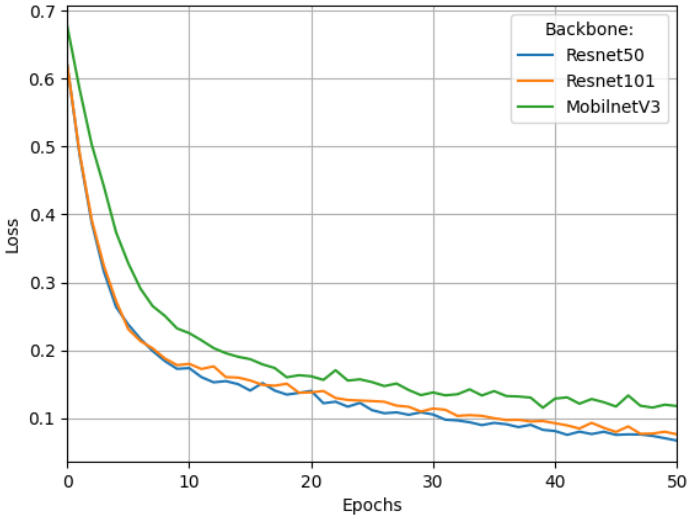


Figure 3.2: The average loss during training for the three feature extractor backbones of the segmentation CNN considered.

During training, we used the Adam optimizer [23], which is a popular optimization algorithm for deep learning. We also used binary cross entropy loss with a sigmoid layer, which is a common loss function for binary classification tasks such as image segmentation (see Figure 3.2). The binary cross entropy loss $\ell(x, y)$ can be defined as

$$\ell(x, y) = L = \{l_1, \ldots, l_N\}^\top, \quad l_n = -w_n \left[y_n \cdot \log \sigma\left(x_n\right) + \left(1 - y_n\right) \cdot \log\left(1 - \sigma\left(x_n\right)\right) \right], \quad (3.1)$$

where x is groundtruth label, y is predicted label, w_n is weight and $\sigma(.)$ is sigmoid layer.

3.3 Evaluation

We evaluated the performance of our CNN for segmenting tree trunks and background using mIoU as the evaluation metric. The Intersection over Union (IoU) measures the overlap between the predicted segmentation (A) and the ground truth (B) labels, and provides a measure of the accuracy of the segmentation task and it is defined as

$$IoU(A, B) = \frac{|A \cap B|}{|A \cup B|}. \quad (3.2)$$

The mIoU metric is simply mean from IoUs computed for each image from testing dataset.

Our experiments involved training the network using three different backbones: ResNet50, ResNet101, and MobileNetV3-Large. After training, we evaluated the performance of each network on a validation set, and computed the mIoU for each backbone.

Our results presented in Table 3.1 showed that the ResNet50 and ResNet101 backbones achieved the highest mIoU of 0.92. The MobileNetV3-Large backbone achieved a lower mIoU of 0.88. These results suggest that the ResNet50 and ResNet101 backbones are more suitable

(a) Example picture

(b) Example picture with predicted mask

(c) Example picture

(d) Example picture with predicted mask

Figure 3.3: Example CNN predictions on pictures from dataset. Fig. 3.3a and Fig. 3.3c are the original, the Fig. 3.3b and Fig. 3.3d are the predictions.

Backbone	mIoU on testing data	mIoU on training data
ResNet50	0.92	0.95
ResNet101	0.92	0.94
MobilNetV3-Large	0.88	0.92

Table 3.1: Comparison of mIoUs for each backbone.

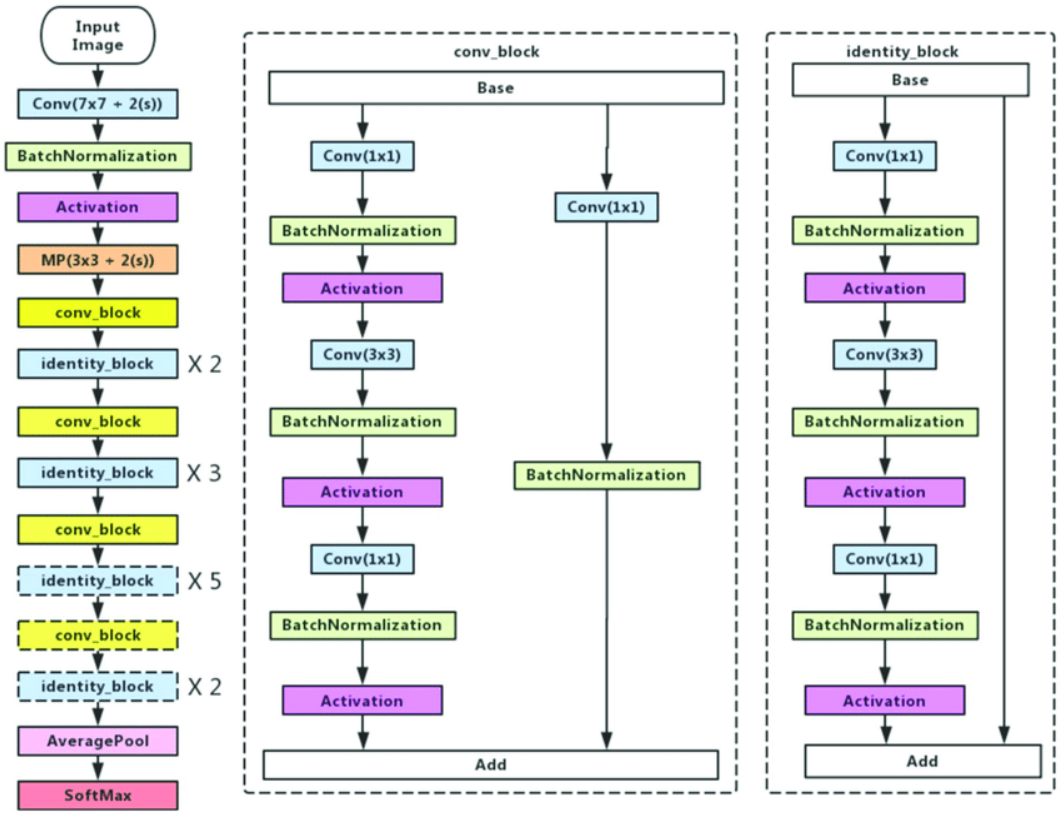


Figure 3.4: A schema of the ResNet50 architecture [14].

for our task of segmenting tree trunks and background, compared to the MobileNetV3-Large backbone. The examples of segmentation are in Figure 3.3.

We selected the ResNet50 backbone (depicted in Figure 3.4) as our final model for several reasons. First, it achieved the highest mIoU among all the tested backbones, indicating that it was able to accurately segment tree trunks and background. Second, it had a lighter architecture compared to the ResNet101 backbone, which makes it more suitable for deployment on devices with limited computational resources.

Chapter 4

Hole Detection

To detect small holes created by bark-beetles in Norway spruce's bark, two algorithms are compared. We gather infected tree samples for testing our solution, however we were not able to gather enough data to experiment with machine learning algorithms at first. Later enough data was obtained to train an object detection network. Our hole detection algorithm is based on the MSER "blob" detector [30] and a further filtration of suspected regions.

4.1 Datasets

For tuning the detectors, samples of infected logs were selected, and pictures were taken using the camera mounted on the UAV. The pictures were taken under various lighting conditions to imitate a range of conditions encountered in the field. The holes were manually annotated by marking the location and size of each hole in the pictures using Datatorch.

Only 98 images of tree bark were acquired from the log samples in the lab. Example images are shown in Figure 4.1. An extension to the dataset was acquired during the MRS experimental campaign in the second half of April in Temešvár, located in the South Bohemian region, which allowed us to gather enough data for training a machine learning model.

The completed dataset is divided into three subsets: training, validation, and testing. Out of the 1,092 images, 874 were allocated for training, 109 for validation, and another 109 for testing to validate the accuracy of detection methods for identifying small holes caused by bark beetles in the tree bark.

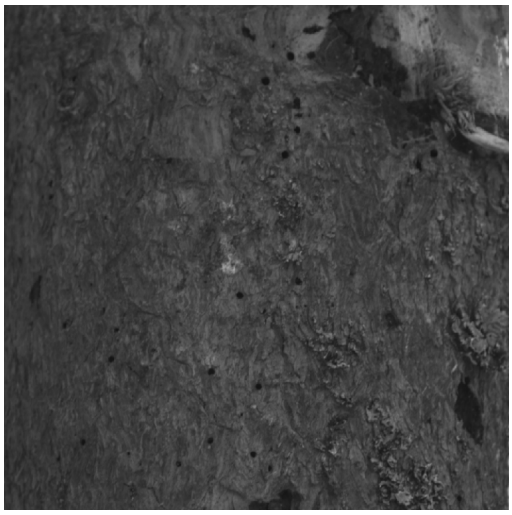
4.2 Hole Detection Algorithm

To compensate various lighting conditions histogram equalization is applied to the image. Then the "blobs", i.e. convex regions with similar intensity, are detected in the picture with the MSER detector. Aftewards, suspected regions are filtered by circularity and intensity. The overview of this process can be seen in Figure 4.2

4.2.1 Histogram Equalization

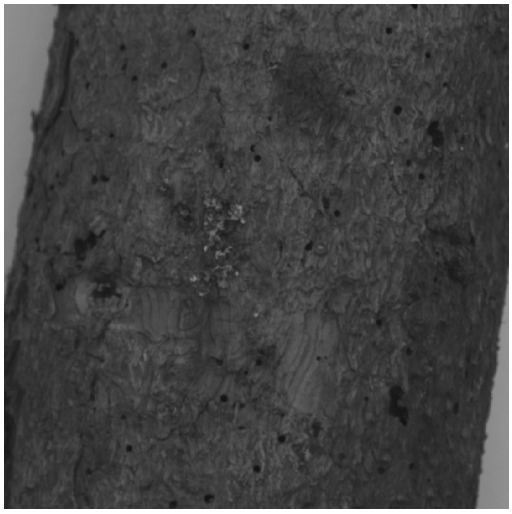
The captured images of the tree bark are converted from RGB to grayscale. To enhance the contrast of the images, we applied an intensity histogram equalization technique. Specifically, CLAHE [31]. This conversion helps to reduce the influence of sharp sunlight in the images, which can obscure or distort the features of interest.

The first step in CLAHE algorithm is to divide the image into tiles. The size of tiles is in our case 8 by 8 pixels and it is one of the two parameters of CLAHE. Histogram equalization

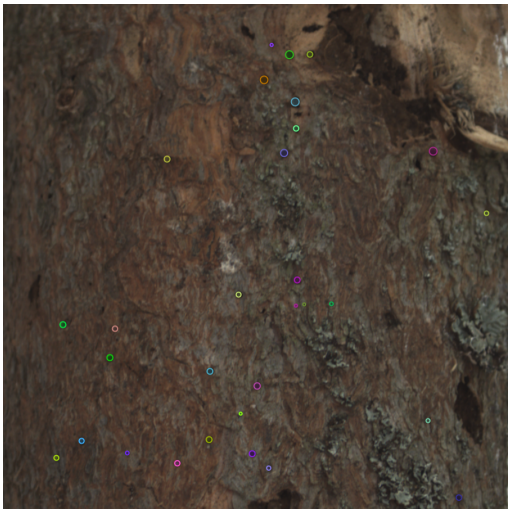


(a) Example picture.

(b) Example picture with annotated holes.



(c) Example picture.



(d) Example picture with annotated holes.

Figure 4.1: Example pictures from the hole detection dataset. Fig. 4.1a and Fig. 4.1c are the original, Fig. 4.1b and Fig. 4.1d are annotated.

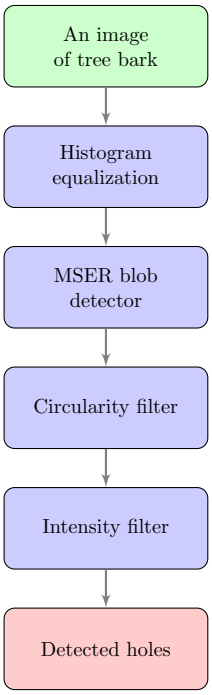


Figure 4.2: Flowchart of the hole detector algorithm.

is applied for each tile. Let us consider the grayscale patch x and let n_i be the number of occurences of gray level i. The probability of an occurence of a pixel of level i in the image is estimated

$$p_x(i) = p(x = i) = \frac{n_i}{n}, \quad 0 \leq i < 256, \tag{4.1}$$

where n is the total number of pixels in the tile and $p_x(i)$ is the tile's normalized histogram for pixel value i.

The cumulative distribution function corresponding to i can be defined as

$$\mathrm{cdf}_x(i) = \sum_{j=0}^{i} p_x(x = j). \tag{4.2}$$

The cumulative distribution function can be interpreted as the tile's accumulated normalized histogram.

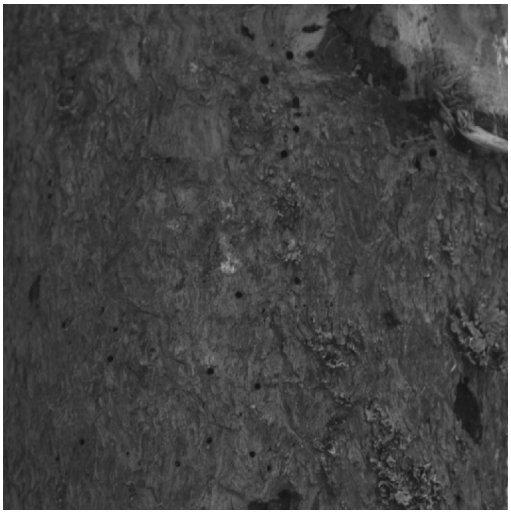
Histogram equalization can be defined as a transformation where the output image has a flat histogram, i e. a linearized cumulative distribution function. The transformation is implemented as

$$y = \mathrm{cdf}_x(k), \tag{4.3}$$

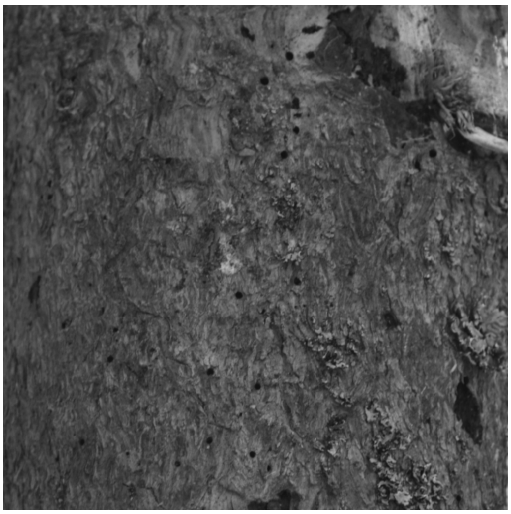
where k is in range $[0, 256]$. The transformation maps the output in the range $[0, 1]$. In order to map the values back to their original range, the following transformation is applied

$$y' = y \cdot (\max\{x\} - \min\{x\}) + \min\{x\} = y \cdot (255). \tag{4.4}$$

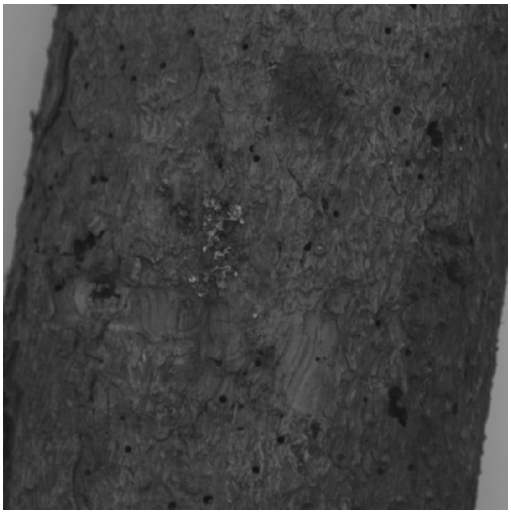
If the contrast is above limit in the equalized histogram, it is clipped. After each tile's histogram is normalized, neighborhooding tiles are blended using bilinear interpolation. An example of a histogram before and after applying CLAHE is in Figure 4.4.



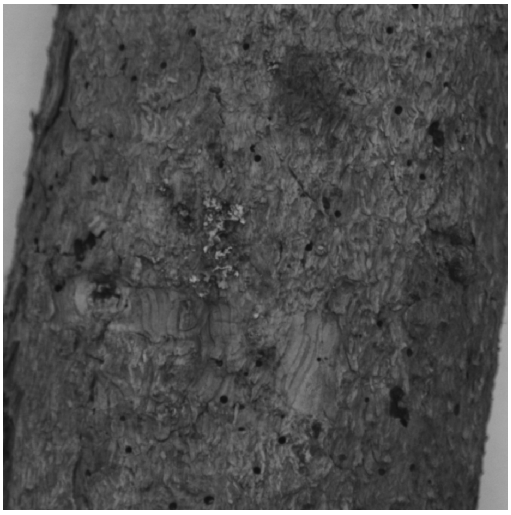
(a) Example picture before applying Contrast Limited Adaptive Histogram Equalization (CLAHE).



(b) Example picture after applying CLAHE.



(c) Example picture before applying CLAHE.



(d) Example picture after applying CLAHE.

Figure 4.3: Example of histogram equalization on pictures from the hole detection dataset. Fig. 4.3a and Fig. 4.3c are the original, Fig. 4.3b and Fig. 4.3d are after CLAHE is applied.

This step improved the accuracy of the detection algorithm by enhancing the contrast of the image while retaining the important features of the tree bark (see Figure 4.3). The contrast limit value was set to a low level of 0.8 to ensure that the shadows in the bark structures were not falsely detected as holes.

4.2.2 MSER Blob Detection

The next step in our detection algorithm utilizes the MSER algorithm to identify blobs in the picture. It is based on the concept of extremal regions, which are connected components of the image that share a common gray-level intensity.

At first, a wide range of thresholds is applied to the image as illustrated in Figure 4.6.

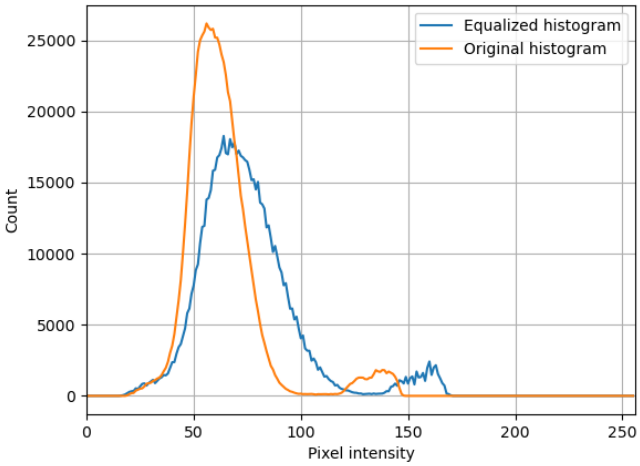


Figure 4.4: An example of an image histogram before and after equalization.

All the pixels below a given treshold are considered as white and all those above are black. As the treshold grows, white spots merge until the whole image is white. The set of all connected components in the sequence is the set of all extremal reagions. The term extremal means that all pixels inside the region have either higher or lower intensity than all the other pixels. An extremal region is deemed "maximally stable" in the local minimum as

$$\mathcal{Q}_{min} = \min\left(\left|\mathcal{Q}_{i+\Delta} \backslash \mathcal{Q}_{i-\Delta}\right| / \left|\mathcal{Q}_i\right|\right), \tag{4.5}$$

where $\mathcal{Q}_i$ is intensity and i is threshold step. These regions are then further filtered based on minimal and maximal area and stability. An example of regions detected by MSER on our dataset is in the Figure 4.5.

4.2.3 Circularity Filter

Based on the assumption that holes in the bark caused by the bark beetle are circular, we implement a circularity filter to eliminate non-circular blobs that correspond to shadows and other false positive detections.

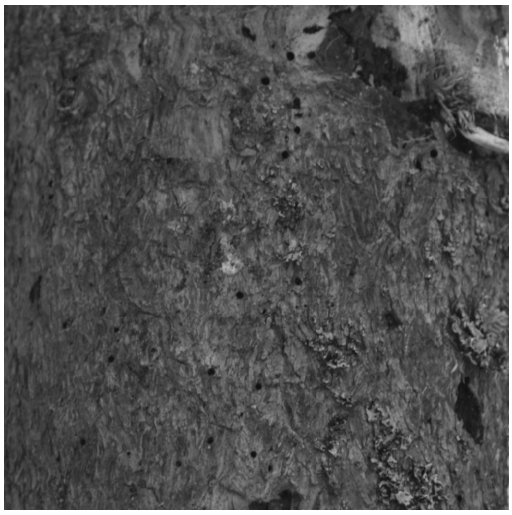
A circularity coefficient c for each detected blob is calculated by comparing the radii of the inscribed circle r_i and the enclosed circle r_e of the detected blob as

$$c = \frac{r_i}{r_e}. \tag{4.6}$$

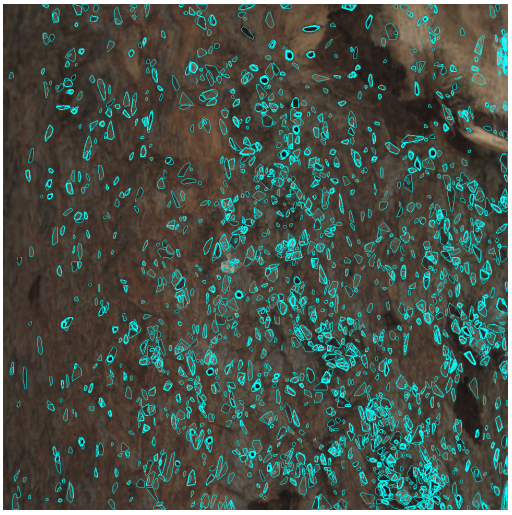
The inscribed circle is the largest circle that can fit entirely inside the detected blob, while the enclosed circle is the smallest circle that completely encompasses the blob.

By calculating the ratio of the radii of the inscribed and enclosed circles, the circularity coefficient of the blob is determined. This coefficient provides a measure of how closely the detected blob resembles a perfect circle. By applying a circularity threshold, we exclude blobs that fall below a certain coefficient value, ensuring that only circular blobs that closely resemble circular holes are retained in the final detection results (see Figure 4.7).

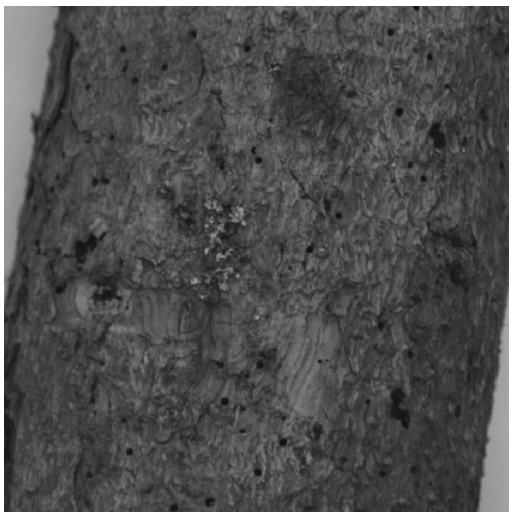
[1]Source: `https://www.micc.unifi.it/delbimbo/wp-content/uploads/2011/03/slide_corso/A34MSER.pdf`



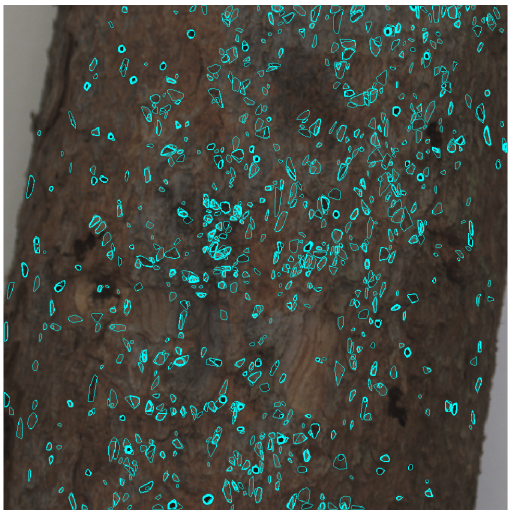
(a) Example picture after applying CLAHE.



(b) Example picture with regions detected by MSER.



(c) Example picture after applying CLAHE.



(d) Example picture with regions detected by MSER.

Figure 4.5: Example of MSER detection. Fig. 4.5a and Fig. 4.5c are after applying CLAHE, Fig. 4.5b and Fig. 4.5d show regions detected by MSER.

4.2.4 Intensity Filters

The final step of the detection algorithm is a series of intensity filters designed to further improve the accuracy of the detection results. These filters eliminate any non-dark blobs detected in the previous steps. This relies on the assumption that true holes are darker than their surrounding.

First, an ellipse is fitted on the blob and inflated by 15 pixels. The difference between inflated ellipse and the blob is the neighborhood that is considered. The mean intensity inside the detected blob i_i is compared with the mean intensity of the blob's surrounding area i_o as follows

$$i_o - i_i > \text{treshold}. \tag{4.7}$$

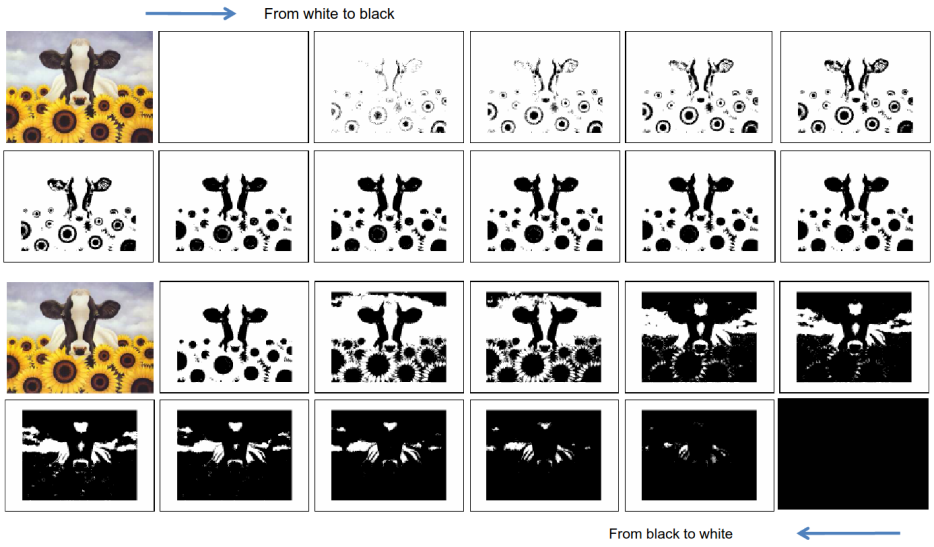


Figure 4.6: An example of thresholding during MSER detection algorithm[1].

If the difference between the neighborhood intensity mean and blob intensity mean is below treshold, the blob is filtered out.

Second, histograms of the intensity values inside I and outside O the detected blob are calculated and their intersection I_{int} is defined as

$$I_{int} = \frac{\sum_{j=1}^{256} \min(I_j, O_j)}{\sum_{j=1}^{256} O_j}, \tag{4.8}$$

where I_j is jth bin of histogram I and O_j is jth bin of histogram O.

The intersection of these histograms is compared to determine whether the intensity inside the blob is significantly darker than the surrounding area. If the area of the intersection is below a certain threshold, the blob is excluded.

The third filter compares the mean intensity inside the detected blob i_i to the mean intensity of the entire image i_p. If the mean intensity of the blob is not lower than 30 % of the mean intensity of the whole image, the blob is excluded. The filter is defined as

$$i_i < 0.3 i_p. \tag{4.9}$$

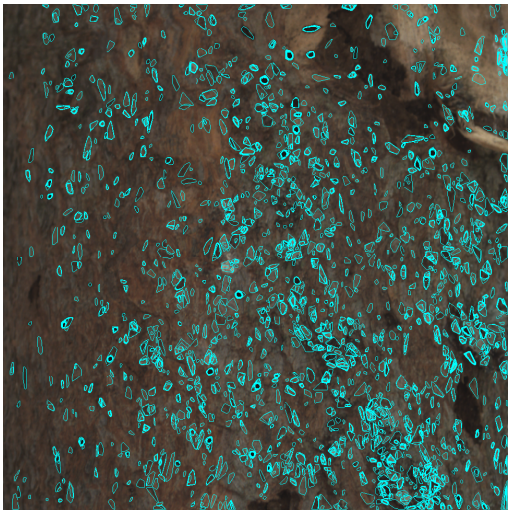
By applying these intensity filters, only truly dark blobs that closely resemble holes caused by bark beetles in the tree bark are retained in the final detection results.

4.3 Evaluation

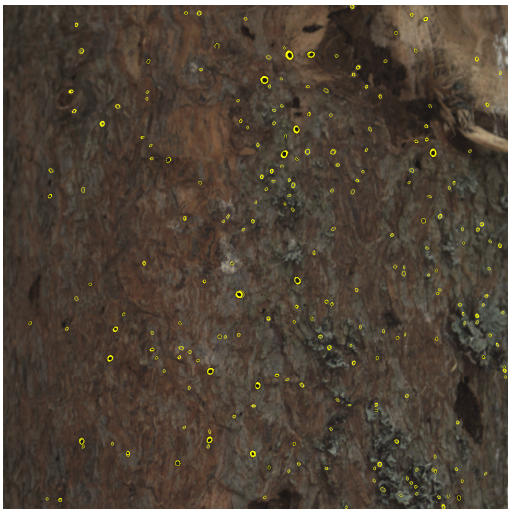
Performance of the detection algorithm on the testing dataset was evaluated using the following metrics:

$$\text{Precision} = \frac{\text{True positives}}{\text{True positives} + \text{False positives}}, \tag{4.10}$$

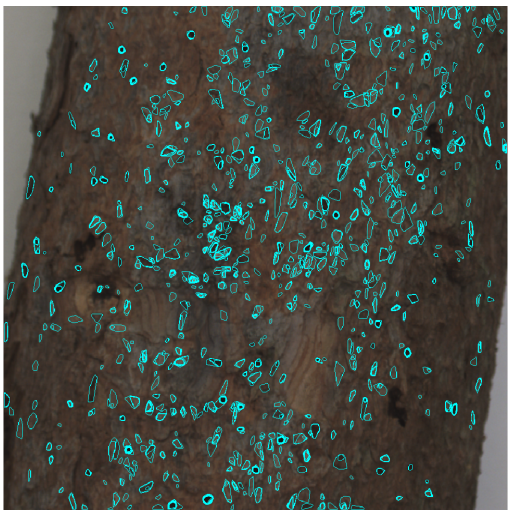
$$\text{Recall} = \frac{\text{True positives}}{\text{True positives} + \text{False negatives}}, \tag{4.11}$$



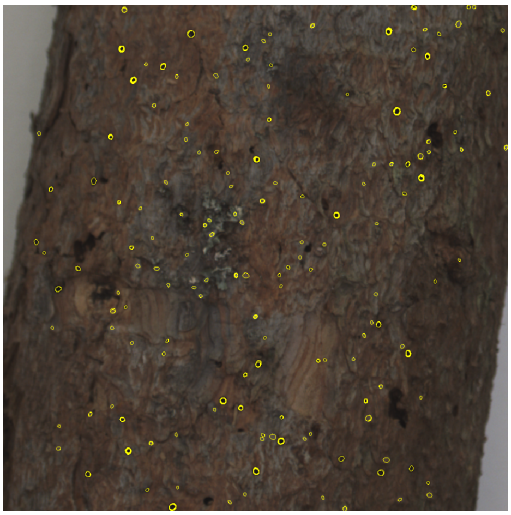
(a) Example picture with regions detected by MSER.



(b) Example picture with regions filtered by circularity.



(c) Example picture with regions detected by MSER.



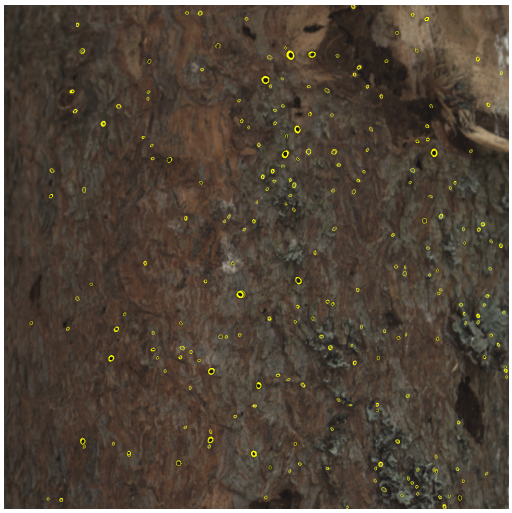
(d) Example picture with regions filtered by circularity.

Figure 4.7: Example pictures before and after applying circularity filter. Fig. 4.7a and Fig. 4.7c show regions detected by MSER,Fig. 4.7b and Fig. 4.7d show regions filtered by circularity.

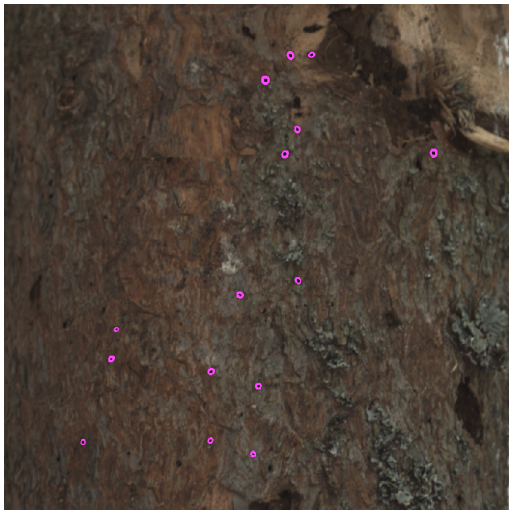
and

$$\mathrm{F1} = \frac{2 \cdot \mathrm{Precision} \cdot \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}}, \tag{4.12}$$

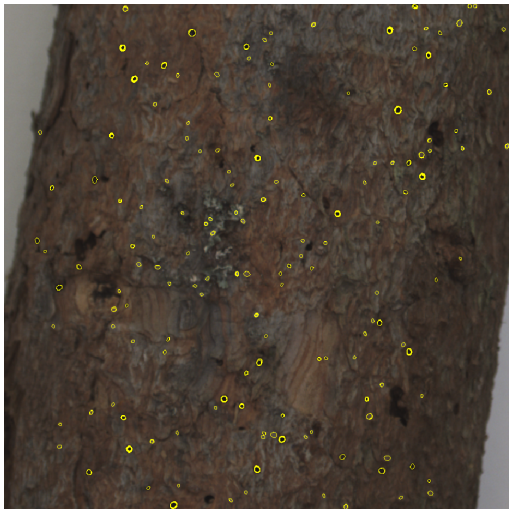
were used. The priority for our evaluation is precision, as we aim to minimize the number of false positives in our detection results. When the tree is infected, there are many holes in the tree bark, but when the tree is healthy there are none. Therefore, it is not required that the detector locates all the holes. If false positive detections are minimized, it is distinguishable whether the bark has holes or not. The results are in Table 4.1.



(a) Example picture with blobs after circularity filter.



(b) Example picture with blobs after intensity filters.



(c) Example picture with blobs after circularity filter.

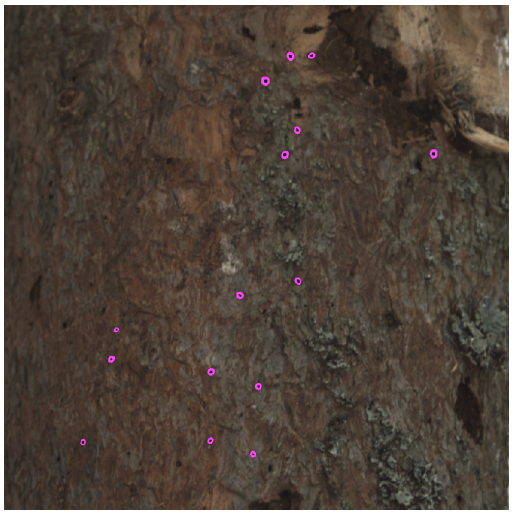
(d) Example picture with blobs after intensity filters.

Figure 4.8: Example of the circularity and intensity filters. Fig. 4.8a and Fig. 4.8c are images with blobs after circularity filter, Fig. 4.8b and Fig. 4.8d are images with blobs after intensity filter was applied.

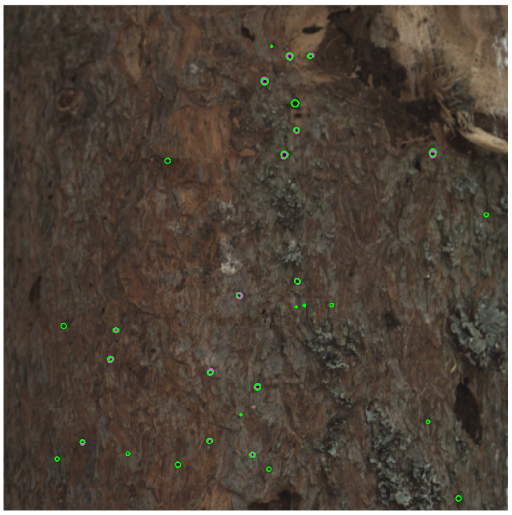
4.4 Comparison with YOLOv7

After extending the dataset, a machine learning model, YOLOv7, can be trained. For training, code and pre-trained weights from [4] are utilized. After training YOLOv7, the hole detection algorithm based on MSER is tested on the same testing dataset. Table 4.2 summarizes the performance of both algorithms on the 109 testing images.

The MSER-based detector requires reconfiguration of its parameters to improve its performance. Although it demonstrated satisfactory results when applied to images captured in lab conditions, it did not perform well when tested on real forest images. This suggests that the MSER detector may not be well-suited for such challenging environmental conditions

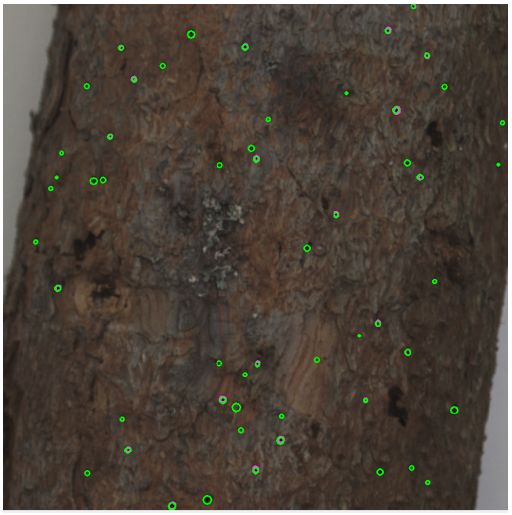


(a) Example picture with detected holes.



(b) Example picture with annotated holes.

(c) Example picture with detected holes.



(d) Example picture with annotated holes.

Figure 4.9: Example output of MSER based hole detector. Fig. 4.9a and Fig. 4.9c have only detected holes drawn (magenta), Fig. 4.9b and Fig. 4.9d have additionally annotated holes drawn (green).

without adjustments.

On the other hand, YOLOv7 exhibited promising results; however, training YOLO effectively requires a substantial amount of data. Unfortunately, obtaining large amounts of data may not always be feasible in practical scenarios, which limits the effectiveness of YOLOv7. Its reliance on a significant volume of training data may pose challenges in real-world situations where obtaining extensive datasets can be difficult. Example detections are in Figure 4.10.

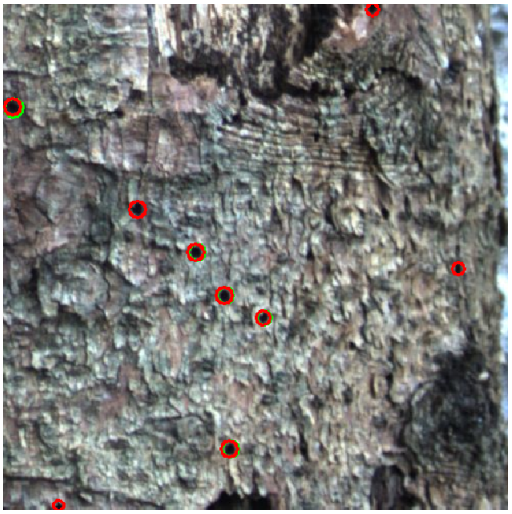
Metric	Value
Precision	0.68
Recall	0.62
F1 score	0.65

Table 4.1: Results of the detector's evaluation on lab dataset.

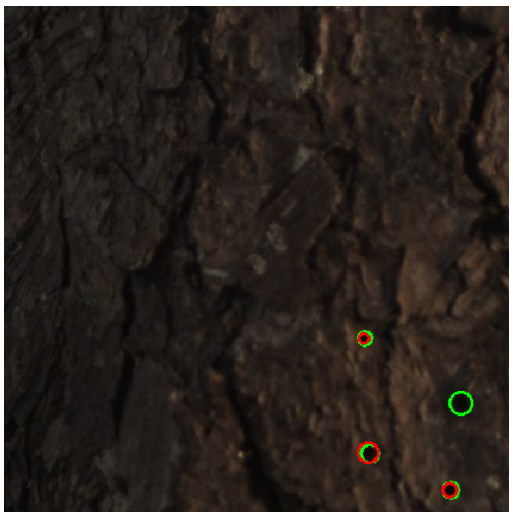
Metric	YOLOv7	MSER-based reconfigured	MSER-based lab config
Precision	0.67	0.58	0.48
Recall	0.71	0.41	0.38
F1 score	0.69	0.48	0.43

Table 4.2: Results of both detectors' evaluation on the new testing set.

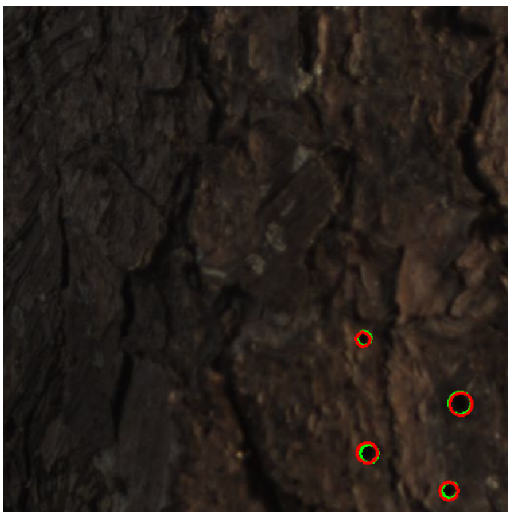
(a) Example picture with detected holes by MSER.



(b) Example picture with detected holes by YOLOv7.



(c) Example picture with detected holes by MSER.



(d) Example picture with with detected holes by YOLOv7.

Figure 4.10: Example output of MSER-based and YOLOv7 hole detection. Fig. 4.10a and Fig. 4.10c show holes detected by MSER, Fig. 4.10b and Fig. 4.10d show holes detected by YOLOv7. Green circles are annotated holes and red circles are detections.

Chapter 5

Tree Classification

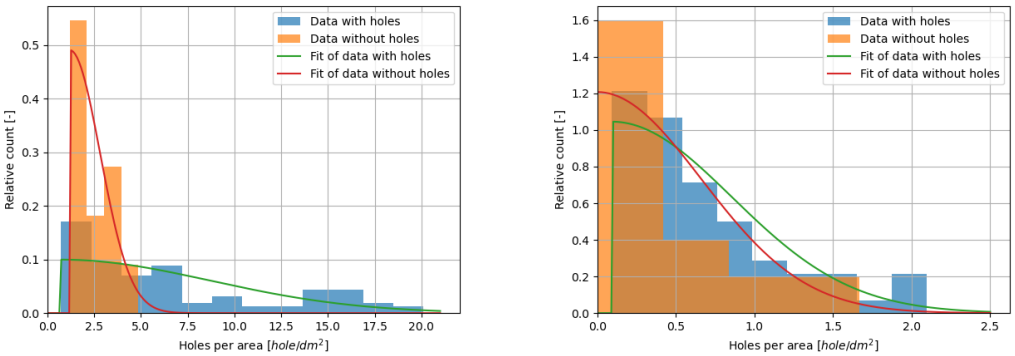
The last step of the pipeline is to determine if the tree is healthy or infected. A confidence measure if the tree is infected is implemented. A set of spruce trees is prepared to test the whole pipeline.

5.1 Dataset

For the evaluation of the system, a dataset was collected during the MRS experimental campaign. The campaign provided an ideal opportunity to test the whole pipeline for detecting bark-beetle infestation, from image acquisition to analysis.

The collected data consisted of high-resolution images of three spruce trees infected with bark-beetle and eight healthy spruce trees. The chosen trees varied in age and size, providing a representative sample of the considered domain. Due to changing weather and surrounding greenery, the lighting conditions during data collection varied, which impacts the accuracy of the detection.

5.2 Confidence Measure



(a) Histogram from data processed by MSER based detector.

(b) Histogram from data processed by YOLOv7 detector.

Figure 5.1: Histograms of holes per area for infected and healthy trees. 5.1a shows results of the MSER-based detector and 5.1b shows results of the YOLOv7 detector.

We propose a new metric to evaluate whether a tree is infected or not. We use data from LiDAR to determine the distance between the camera and the tree, which allows us to

			MSER-based detector				YOLOv7			
ID	imgs	State	hpa	p_h	p_i	$\boldsymbol{CM}$	hpa	p_h	p_i	$\boldsymbol{CM}$
0	6	Infected	17.57	0.000	0.010	**1.000**	3.52	8.4e−7	4.4e−5	**0.981**
1	5	Infected	1.62	0.474	0.099	**0.173**	0.12	1.187	1.045	**0.468**
2	5	Infected	5.51	0.014	0.083	**0.853**	1.70	0.044	0.114	**0.720**
3	6	Healthy	1.26	0.490	0.100	**0.169**	0.03	1.206	0.000	**0.000**
4	7	Healthy	0.98	0.000	0.100	**0.091**	0.01	1.207	0.000	**0.000**
5	7	Healthy	1.37	0.488	0.100	**0.169**	0.02	1.207	0.000	**0.000**
6	4	Healthy	0.45	1.000	0.000	**0.000**	0.08	1.198	0.000	**0.000**
7	6	Healthy	2.98	0.268	0.100	**0.268**	0.12	1.187	1.045	**0.468**
8	5	Healthy	1.39	0.487	0.099	**0.169**	0.21	1.148	1.033	**0.474**
9	8	Healthy	1.28	0.490	0.100	**0.168**	0.02	1.207	0.000	**0.000**
10	8	Healthy	1.63	0.473	0.099	**0.173**	0.02	1.207	0.000	**0.000**

Table 5.1: Results of the whole pipeline for every testing tree with both hole detectors.

calculate the area of the tree's bark that has been scanned based on the tree bark segment and known parameters of the camera as

$$a = \frac{0.5 \cdot \pi \cdot d^2 \cdot w \cdot h}{f_x \cdot f_y}, \tag{5.1}$$

where a is the area, d is the distance of the camera to the tree, w is width of the tree trunk in pixels, h is heigth of the tree trunk in pixels and f_x, f_y are focal lengths in vertical and horizontal directions. We then count the number of detected holes per area, which gives us an indication of the severity of the bark beetle infestation.

To calculate the probability of infection for a given number of holes per area, we created a normalized histogram using the testing dataset from the lab samples. Each histogram is then fitted with probability density function of folded normal distribution. Histograms and probability density functions are shown in Figure 5.1. The fitted distribution may be shifted along the x axis, which means that values lower than the histogram are qualified as zero.

If a tree has n holes per area, the final confidence measure CM is computed

$$CM = \frac{p_{infected}}{p_{infected} + p_{healthy}}, \tag{5.2}$$

for given n holes per area, where $p_{infected}$ is probability of tree being infected and $p_{healthy}$ is probability of tree being healthy. The probabilities are acquired from estimated folded normal distribution. This provides us with a probability estimate of whether a tree is infected or not based on the number of detected holes per area.

5.3 Results

We test the classification pipeline on eleven trees, out of which three are infected. The results are in Table 5.1. Both algorithms for hole detection, MSER based and YOLOv7, are tested for this purpose.

For two out of three infected trees, the MSER-based algorithm correctly identifies them with 100 % and 85 % confidence. However, one of the infected trees is falsely evaluated with

17 % confidence by the whole pipeline. For the healthy trees, the maximum confidence is a 26.8 %. The pipeline with YOLOv7 as hole detector similarily identified two infected trees. The falsely classified infected tree has 46.8 % confidence, but so does two healthy trees.

The falsely evaluated trees were photographed under direct sunlight, which caused problems and confusion in the pipeline for both hole detectors. Overall, the results demonstrate that our algorithm has a high accuracy rate for identifying infected trees, with some limitations in certain lighting conditions.

Chapter 6

Conclusion

The objective of this thesis was to design a system using an UAV to detect trees infected with the European spruce bark beetle. The UAV was deployed beneath tree canopies to scan the tree trunks within areas suspected of bark beetle infestation. The system relies on inputs from a high-resolution RGB camera to capture images, and the distance from the camera to the tree is obtained through a LiDAR sensor.

The output of the system is a confidence measure indicating whether a tree is infected or not. To achieve this, we divided the system pipeline into several stages. The first step involves segmenting the tree trunks from the RGB images. We created a labeled dataset and trained a ResNet50 segmentation network.

The second part of the pipeline is hole detection within the segmented tree trunks. Due to a limited amount of data available, we propose an algorithm based on MSER blob detection. The MSER-based detector employes histogram equalization on the images, detects blobs using the MSER algorithm, and then filters them based on circularity and intensity properties. Initial results obtained using the MSER-based detector showed promise when tested on laboratory data.

During an experimental campaign, we collected additional data, enabling us to create a new dataset with labeled bark beetle-made holes. This dataset is utilized to train the YOLOv7 model, and its performance exceeds that of the MSER-based detector when applied to real-world data.

The final step of the pipeline involves evaluating whether a tree is infected or not. We introduce a metric called “holes per area”, which represents the number of detected holes relative to the visible bark area captured in the image. This metric is then compared with histograms, which are fitted with a probability density of fold normal distribution, of healthy and infected trees to make the infection determination. Even though YOLOv7 is better hole detector than MSER-based detector, it achieves similar results when connected in the pipeline.

By combining these stages our system successfully detected trees infected with the European spruce bark beetle, providing a valuable tool for early detection and monitoring in forest ecosystems.

6.1 Future work

Next steps for this project involve expanding and improving the datasets used in all stages of the system pipeline. This includes collecting more data for tree trunk segmentation, hole detection, and tree classification. Additionally, efforts will be made to evaluate the system on a wider range of tree samples to ensure its effectiveness in different environments.

Another area of focus is the refinement of the histograms used for tree classification. Research will be conducted to explore alternative probability density functions that can better represent the distribution of healthy and infected trees. This will involve calculating new histograms based on the extended dataset and finding the best-fitting functions to improve the accuracy of the infection determination process.

Further improvements can be made in the hole detection stage. One potential avenue is to investigate the combination of blob filtering techniques with the YOLOv7 detection model. By integrating these approaches, it is expected that more accurate and reliable hole detection results can be achieved. Additionally, exploring the integration of YOLOv7 results with tree classification is important, as despite the advancements in hole detection, the overall improvement in tree classification compared to the MSER-based detector needs to be further investigated.

Chapter 7

References

[1] M. Petrlik, P. Petracek, V. Kratky, *et al.*, "UAVs Beneath the Surface: Cooperative Autonomy for Subterranean Search and Rescue in DARPA SubT," *Field Robotics*, vol. 3, pp. 1–68, 2023. DOI: https://doi.org/10.55417/fr.2023001.

[2] D. Hert, T. Baca, P. Petracek, *et al.*, "MRS Modular UAV Hardware Platforms for Supporting Research in Real-World Outdoor and Indoor Environments," in *2022 International Conference on Unmanned Aircraft Systems (ICUAS)*, IEEE, 2022, pp. 1264–1273. DOI: 10.1109/ICUAS54217.2022.9836083.

[3] X. Liu, G. V. Nardari, F. C. Ojeda, *et al.*, "Large-scale autonomous flight with real-time semantic slam under dense forest canopy," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 5512–5519, 2022. DOI: 10.1109/LRA.2022.3154047.

[4] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," 2022. arXiv: 2207.02696 [cs.CV].

[5] T. Baca, M. Petrlik, M. Vrba, *et al.*, "The MRS UAV System: Pushing the Frontiers of Reproducible Research, Real-world Deployment, and Education with Autonomous Unmanned Aerial Vehicles," *Journal of Intelligent & Robotic Systems*, vol. 102, no. 26, pp. 1–28, 1 May 2021.

[6] V. Bárta, P. Lukeš, and L. Homolová, "Early detection of bark beetle infestation in norway spruce forests of central europe using sentinel-2," *International Journal of Applied Earth Observation and Geoinformation*, vol. 100, p. 102 335, 2021, ISSN: 1569-8432. DOI: https://doi.org/10.1016/j.jag.2021.102335. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0303243421000428.

[7] F. M. Hellwig, M. A. Stelmaszczuk-Górska, C. Dubois, *et al.*, "Mapping european spruce bark beetle infestation at its early phase using gyrocopter-mounted hyperspectral data and field measurements," *Remote Sensing*, vol. 13, no. 22, 2021, ISSN: 2072-4292. DOI: 10.3390/rs13224659. [Online]. Available: https://www.mdpi.com/2072-4292/13/22/4659.

[8] T. Hlásný, K. Merganičová, R. Modlinger, R. Marušák, R. Löwe, and M. Turčáni, "Prognosis of bark beetle outbreak and a new platform for the dissemination of information about the forests in the czech republic," *ZPRÁVY LESNICKÉHO VÝZKUMU*, vol. 66, pp. 197–205, 2021. DOI: https://www.vulhm.cz/files/uploads/2021/09/637.pdf.

[9] D. S. Jodas, S. Brazolin, T. Yojo, *et al.*, "A deep learning-based approach for tree trunk segmentation," in *2021 34th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, 2021, pp. 370–377. DOI: 10.1109/SIBGRAPI54419.2021.00057.

[10] L. Liu, A. Zhang, S. Xiao, *et al.*, "Single tree segmentation and diameter at breast height estimation with mobile lidar," *IEEE Access*, vol. 9, pp. 24 314–24 325, 2021. DOI: 10.1109/ACCESS.2021.3056877.

[11] R. Minařík, J. Langhammer, and T. Lendzioch, "Detection of bark beetle disturbance at tree level using uas multispectral imagery and deep learning," 2021. DOI: 10.3390/rs13234768. [Online]. Available: https://www.mdpi.com/2072-4292/13/23/4768.

[12] X. Yanzhe, W. Teresa, G. Fei, C. Jennifer, and B. Kevin, "Improved small blob detection in 3d images using jointly constrained deep learning and hessian analysis," 2020. DOI: https://doi.org/10.1038/s41598-019-57223-y.

[13] A. Howard, M. Sandler, G. Chu, *et al.*, "Searching for mobilenetv3," 2019. arXiv: 1905.02244 [cs.CV].

[14] Q. Ji, J. Huang, W. He, and Y. Sun, "Optimized deep convolutional neural networks for identification of macular diseases from optical coherence tomography images," *Algorithms*, vol. 12, p. 51, Feb. 2019. DOI: 10.3390/a12030051.

[15] A. Paszke, S. Gross, F. Massa, *et al.*, "PyTorch: An Imperative Style, High-Performance Deep Learning Library," arXiv:1912.01703 [cs, stat], Dec. 2019. DOI: 10.48550/arXiv.1912.01703. [Online]. Available: http://arxiv.org/abs/1912.01703.

[16] J. Tordesillas, B. T. Lopez, and J. P. How, "Faster: Fast and safe trajectory planner for flights in unknown environments," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 1934–1940. DOI: 10.1109/IROS40897.2019.8968021.

[17] H. Abdullah, R. Darvishzadeh, A. K. Skidmore, T. A. Groen, and M. Heurich, "European spruce bark beetle (ips typographus, l.) green attack affects foliar reflectance and biochemical properties," *International Journal of Applied Earth Observation and Geoinformation*, vol. 64, pp. 199–209, 2018, ISSN: 1569-8432. DOI: https://doi.org/10.1016/j.jag.2017.09.009. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0303243417301988.

[18] J. Lubojacký, M. Knížek, and J. Liška, "Symptomy napadení stromů kůrovci ve smrkových porostech," 2018, Výzkumný ústav lesního hospodářství a myslivosti, v. v. i.

[19] Y. Majeed, J. Zhang, X. Zhang, *et al.*, "Apple tree trunk and branch segmentation for automatic trellis training using convolutional neural network based semantic segmentation," *IFAC-PapersOnLine*, vol. 51, no. 17, pp. 75–80, 2018, 6th IFAC Conference on Bio-Robotics BIOROBOTICS 2018, ISSN: 2405-8963. DOI: https://doi.org/10.1016/j.ifacol.2018.08.064. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405896318311807.

[20] K. Mohta, K. Sun, S. Liu, *et al.*, "Experiments in fast, autonomous, gps-denied quadrotor flight," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 7832–7839. DOI: 10.1109/ICRA.2018.8463214.

[21] L. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *CoRR*, vol. abs/1706.05587, 2017. arXiv: 1706.05587. [Online]. Available: http://arxiv.org/abs/1706.05587.

[22] C. Eggert, S. Brehm, A. Winschel, D. Zecha, and R. Lienhart, "A closer look: Small object detection in faster r-cnn," in *2017 IEEE International Conference on Multimedia and Expo (ICME)*, 2017, pp. 421–426. DOI: 10.1109/ICME.2017.8019550.

[23] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2017. arXiv: 1412.6980 [cs.LG].

[24] M. Zhang, T. Wu, S. C. Beeman, *et al.*, "Efficient small blob detection based on local convexity, intensity and shape information," *IEEE Transactions on Medical Imaging*, vol. 35, no. 4, pp. 1127–1137, 2016. DOI: 10.1109/TMI.2015.2509463.

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015. arXiv: 1512.03385 [cs.CV].

[26] T.-Y. Lin, M. Maire, S. Belongie, *et al.*, "Microsoft coco: Common objects in context," 2015. arXiv: 1405.0312 [cs.CV].

[27] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," 2015. arXiv: 1505.04597 [cs.CV].

[28] H. Kong, H. C. Akakin, and S. E. Sarma, "A generalized laplacian of gaussian filter for blob detection and its applications," *IEEE Transactions on Cybernetics*, vol. 43, no. 6, pp. 1719–1733, 2013. DOI: 10.1109/TSMCB.2012.2228639.

[29] D. Nistér and H. Stewénius, "Linear time maximally stable extremal regions," in *Computer Vision – ECCV 2008*, D. Forsyth, P. Torr, and A. Zisserman, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 183–196, ISBN: 978-3-540-88688-4.

[30] J Matas, O Chum, M Urban, and T Pajdla, "Robust wide-baseline stereo from maximally stable extremal regions," *Image and Vision Computing*, vol. 22, no. 10, pp. 761–767, 2004, British Machine Vision Computing 2002, ISSN: 0262-8856. DOI: https://doi.org/10.1016/j.imavis.2004.02.006. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0262885604000435.

[31] S. M. Pizer, E. P. Amburn, J. D. Austin, *et al.*, "Adaptive histogram equalization and its variations," *Computer Vision, Graphics, and Image Processing*, vol. 39, no. 3, pp. 355–368, 1987, ISSN: 0734-189X. DOI: https://doi.org/10.1016/S0734-189X(87)80186-X. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0734189X8780186X.

Chapter A

Appendix A

A list of all attached files is in Table A.1.

Filename	Comment
`config_real_data.yml`	Config file for MSER-based hole detector.
`download.txt`	Link to cloud storage with datasets and segmentation model.
`hole_detector.py`	MSER-based hole detector class.
`imgs_holes_results.npy`	Data for histogram for tree classification.
`imgs_no_holes_results.npy`	Data for histogram for tree classification.
`pipeline_demo.py`	Executable demo of the system with MSER-based detector.
`trunk_segment.py`	Tree trunk segmentation class.
`utils.py`	Helper functions.

Table A.1: A list of attached files.