Tree Localization and Monitoring on Autonomous Drones employing Deep Learning

Lars Fichtel, Alexander M. Frühwald, Leonhard Hösch, Vitaliy Schreibmann, Christian Bachmeir
University of Applied Sciences Würzburg-Schweinfurt
{lars.fichtel, alexander.fruehwald, leonhard.hoesch, vitaliy.schreibmann, christian.bachmeir}@fhws.de

Frank Bohlander
University of Applied Sciences Erfurt
Erfurt, Germany
frank.bohlander@fh-erfurt.de

***Abstract*—Forest management relies on the analysis of satellite imagery and time intensive physical on-site inspections. Both methods are costly and time consuming. Satellite based images are often not updated in a sufficient frequency to react to infestations or other occurring problems.**

Forest management benefits greatly from accurate and recent information about the local forest areas. In order to react appropriately and in time to incidents such as areas damaged by storms, areas infested by bark beetles and decaying ground water level, this information can be extracted from high resolution imagery.

In this work, we propose UAVs to meet this demand and demonstrate that they are fully capable of gathering this information in a cost efficient way. Our work focuses on the cartography of trees to optimize forest-operation. We apply deep learning for image processing as a method to identify and isolate individual trees for GPS tagging and add some additional information such as height and diameter.

I. Introduction

Forestry data for long term planning, such as forest inventory, is still gathered manually or by the analysis of satellite-images. These methods are time consuming and potentially costly, leading to an insufficient update frequency for reacting to infestations or other time-critical events.

The average territory managed by a local forester measures between 2500ha up to 8000ha [1]. Thus, a close observation of the whole area is challenging to do on foot or by ground based vehicles. Forestry inventory and planning could be improved in terms of granularity and time efficiency if this information could be obtained from aerial photos with the aid of an AI. This would enable the forestry to identify not only a single tree, but also the species and monitor the standing volume of the trees.

If this information is required urgently and on a larger scale, for example in the case of vast areas damaged by storms, covered by snow, infested by bark beetles or foliage feeding larvae of different primary forest pests, satellite imagery might be of insufficient resolution [2]. Additionally, satellite imagery might be outdated, as it is usually refreshed periodically. As an example, Landsat 7 covers a quarter of earths landmass every 16 days. Still, depending on cloud and seasonal vegetation changes, a refresh of image data is not guaranteed for any amount of time [3].

Therefore, the usage of drones to collect imagery, combined with AI based assessment methods for data evaluation, will speed up the process. In case of catastrophes, like wildfires or storm damages, high resolution imagery could enable the following options:

- Affected areas can be located quickly.
- Marketable wood volume of trees and reductions through pests or storms can be documented.
- The amount and types of new plants and trees for replanting can be assessed quickly.
- The composition of a forest can be analyzed, which allows to identify and manage the need for enrichment planting.

In case of pests, high resolution imagery could enable the following options:

- Infested trees or tree groups (e.g. by bark beetles) can be detected in time to react. Fettig et al. recommend that potential measures are to be conducted immediately after the detection [4, 5].
- The development of an infestation of leaf and needle feeding larvae can be detected due to damage of foliage or appearance of larvae.
- Areas with need for pesticide application can be identified. AI identification might reduce the area that needs treatment and thus reduces the harm to non-target species and also reduces costs for the treatment.

GPS coordinates allow forest laborers to quickly and accurately find affected trees or areas using smart-phones or GPS devices.

A. *Project Goal and Steps*

Our goal is to enable forest management on an individual tree basis. This allows to recognize stress as well as pests early and gives a forester the possibility to react accordingly.

Our short-term goal is to collect image material of German forests and identify individual trees. This information will be compiled into an data base. On a long term scale, this data base will be extended with other information that can be extracted by AI and collected by drones.

To collect this kind of data, we design an affordable prototype platform, utilizing tree recognition software based on prototypical AI. We use the flexibility of a hovering drone with mounted sensors and cameras for navigation and data collection. Our drone autonomously follows a predefined path above the trees, captures images of the canopy and applies deep learning for localization of individual trees. This approach is scalable and requires minimal human interaction. Human input is only necessary to define the area and to bring the drone to its starting area. With this approach, we are able to obtain GPS coordinates and additional information for every recognized tree, which then can be added to our data set.

With repeated overflights, our drone can recognize individual trees in real-time and can append the current information to the existing data, thus generating a time series of the forest development. Building on this ability, the data can be extended in the future by enriching the tree location with other features such as color, height and approximate age.

B. Related Work

The past years have shown an increased interest in the observation of forests for multiple reasons, may it be to prevent forest fires or to ensure the healthiness of the trees. Most of these observations are conducted using satellites or airplanes, as seen in [6], [7] and [8]. Especially airborne LIDAR has been proven to be useful for natural resource assessment [9].

Still, airborne and satellite-based image acquisition is quite expensive, especially if it has to be conducted repeatedly. It often requires a trade-off between frequency, scale and resolution [10]. Unmanned aerial vehicles, on the other hand, are very cheap, fast and versatile in comparison to satellites and airplanes [10], which is why there is a lot of research regarding the use of drones for this task in the last years:

Sankey et al. successfully used drones for forest monitoring in the southwestern USA [11]. Almeida et al. used a drone mounted LIDAR to monitor the structure of forest restoration plantations, evaluating information like canopy height, gap fraction and leaf area index [12]. They used the collected data to identify the tree density of previously specified areas, but do not provide information for individual trees.

Zhang et al. used drones and a canopy forest model with focus on collecting data for long-term forest monitoring [13]. Their approach does not include any tree detection but focuses on a digital model to extract and comprehend forest data. In contrast, our research maps the whole forest with individual trees for further analysis. Also, Zhang et al. do not provide any data on the individual trees and their location. They focus on the overall forest canopy.

Morsdorf et al. use LIDAR for geometric reconstruction at single tree level for forest wildfire management, which is a use case pretty similar to the one presented in this work [14]. While their results are very accurate, they also state that they can only detect the triangular shape of the trees present in the Ofenpass area in the Swiss National Park. As a result of this, it is not directly applicable for the more diverse German forestry with wider trees that do not fit the triangular shape.

Yue et al. compare different segmentation algorithms to be used on airborne photographs of buildings [15]. While segmentation is also applied to differentiate forests from fields or other areas, it struggles with segmentation of similar and overlapping objects such as trees in a forest. The segmentation also required high quality and granularity of the input data to avoid one pixel representing an entire class.

Kampen et al. achieve a similar goal like this work with airborne multispectral cameras [16]. The main difference between the work of Kampen et al. and the approach presented in this paper is the usage of on-device calculations, a simpler camera and a smaller drone. This leads to massively reduced costs. A camera feed combined with the height sensor of the drone suffices as data source for the given task.

The recent years also showed an increased usage of artificial intelligence to detect single entities in a densely populated area. Zhao et al. reviewed generic object detection with deep learning [17]. In a more practical example, Liu and Wang used deep learning to detect broken corn and Kuo and Lin detected road signs in DVR images with deep learning [19]. Mery et al. detected persons in a crowded room with a simple smart phone camera and the assistance of deep learning [20]. We believe that these methods can be transferred to the task of detecting individual trees in a forest.

II. Defining the Technical System Requirements

A basic autonomous drone system needs a frame with rotors and servos, a flight controller, a battery and sensors. To comply with German and EU regulations, it is a requirement to be able to fly the drone manually in emergency situations. Therefore, we additionally require an RC kit consisting of the RC itself and an antenna on the drone.

A. Drone Requirements

We define following requirements for the sensors mounted on the drone:

- The sensors must provide sufficient data to the flight controller to guarantee a stable flight.
- The provided data must allow the flight controller to fly to predefined coordinates.
- The data also must be sufficient for object detection of anything that can appear in a forest, including agriculture vehicles and allow collision avoidance.
- The exact distance to the ground must be known. Conventional height-maps usually don't account for vegetation or are outdated, which means they are not suited for navigation in the given use-case [21].
- The captured images must be sufficient to identify and map trees.

For evaluating the sensor data and controlling the flight mission, we need to install another computer on board of the drone. The computing unit with CUDA cores, specialized in AI calculations, is tasked with image prepossessing and inference.

In addition to image processing, our Jetson computing unit also manages sensor data evaluation and collision avoidance.

B. AI Requirements

Considering the AI, the most important requirement is the ability to infer the size and location of the trees during flight. At first, a candidate for classification as a tree has to be selected and its location has to be calculated. Especially convolutional neural networks achieved high accuracy with significantly reduced error rate in classification tasks [22]. The development of neural networks for classification and localization of objects enables us to provide robust results in a short period of time [23].

With neural networks, we can identify and locate trees directly from images without the necessity of manual feature extraction for model training [24]. In case of traditional computer vision, it is necessary to create a number of features for every tree species, which is a time consuming task.

Large scale deep learning models with millions of parameters, such as ResNet, have a big memory footprint and require a lot of computing time [25]. As defined in the project goals, the drone has to locate and identify the trees in real-time. This is especially important due to a lack of adequate mobile communication coverage, a continuous connection to GPU cloud computing providers is not ensured above German forests.

Uploading the taken images for later inference would provide disadvantages in operation. As trees are tracked over multiple images, grave errors like blurry images can be detected and counteracted immediately.

Another requirement considering the AI architecture is the number of detected objects per image. For our use-case, the AI has to locate multiple objects of the same type in an image, without being limited by a maximum number of objects.

The project goal requires the AI to identify individual trees in a forest. This task includes the tasks of knowing what kinds of objects exist in the image, as well as where they are located.

Neural Nets with focus on segmentation such as TreeSeg-Net [26] can recognize forest area but would fail to of identify individual trees. These requirements shorten the list of applicable AI architectures significantly, as neural networks such as VGG [27] and AlexNet [28] focus on identifying objects in the image instead of locating them.

The selection of the computing unit is restricted by the chosen AI-Architecture in terms of required memory, type and amount of computing units. Additional requirements are their weight and price. The power requirements of the system are not considered in detail, as the amount of power used is low compared to the amount required for flying.

Additionally, we formulate following requirements for the AI:

- The AI must infer the image in near real-time on our computing unit.
- The AI has a robust model for transfer learning with the ability to identify plants.
- The AI architecture provides capabilities to reduce over- and under-fitting risk.
- The output of the AI consists of bounding boxes and confidence values.
- Training should be possible on partially labeled images.
- The AI should be able to process images from an HD camera.

III. Component Selection

The drone is required to carry all the required sensors for a sufficient amount of time to fulfill the given task. Thus, we decided to utilize an industrial-sized drone frame as a base, which provides space for future extensions as well as a sufficient flight time.

This left us with two options available: Using a bare-bone DJI S1000+ frame [29] or an already complete set up with a DJI M600 Pro [30]. The M600 comes with a A3 Pro flight kit including multiple GPS antennas and a flight controller [31], while the F550 and the S1000+ only consist of the frame, the rotors and the servos. The DJI Flight Controller's API proved to be too inflexible for autonomous flight as it provides limited control over the flight itself. Therefore, we decided to use the S1000+ with a Pixhawk 2.1 Cube [32] that provides the necessary interface to control autonomous flight and 3 IMUs.

Considering the sensors for the closed-loop system controlling the flight, we choose the three redundant on-board IMUs of the Pixhawk Cube.

Next, we chose a Here 2 GNSS to further enhance positioning with a gyrometer, compass, accelerometer and a barometer. This provides global positioning and the correction of possible accumulating errors from the IMUs. The Here 2 GNSS supports the GPS, GLONASS, Galileo and BeiDou positioning systems as well as the satellite based augmentation systems WAAS, EGNOS, MSAS and GAGAN [33]. The Here 2 is able to reach centimetre-level accuracy.

We chose LIDAR systems for collision avoidance, as they have a high range and accuracy while keeping a high update rate. However, fixed LIDAR systems on drones do have disadvantages like reflections, materials that are not detected and a small angle of beam spread. Thus, we decided to additionally use ultrasonic modules to compensate these weaknesses and allow testing and flight in and near urban areas.

We decided to use a CAA L-8 system by EmQopter [34]. This system consists of 8 LIDAR and 12 ultrasonic sensors, which are distributed equally in one plane, as displayed in figure 1. The system is running on an update frequency of 100 hertz.

We added a Garmin LidarLite v3 Sensor to allow for accurate height measurements [35]. For the camera, we decided to go for a Zed mini. It has an inbuilt gyroscope, an accelerometer and two image sensors with 4 Megapixels each. The different relations of resolution and frames per second can be seen in Table I. As the drone is in motion, a high video frame rate may compensate blur.

Fig. 1. CAA L-8 system, the LIDAR system used in the presented example

Resolution	$2.200p$	$1080p$	$720p$	$WVGA$
Frames per Second	15	30	60	120

TABLE I
RESOLUTION COMPARED TO FRAMES PER SECOND

For computing, we utilized a NVIDIA Jetson platform, which supports GPU computing based on CUDA. This allows us to run AI operations on the GPU, which is faster and means less load on the CPU. In addition to Tensorflow, the Jetson also runs frameworks such as ROS, OpenCV, and Matlab. This flexibility is the reason, we decided to add a Jetson TX2 Nano with a Developer board. It brings computing power to process the camera pictures, the collision avoidance and high-level flight planning.

For the remote control, we selected the Taranis X9D [36] as a remote control and the FrSky X8R [37] receiver because of its 1.5km range and 16 channels. We chose 6S lithium polymer batteries with 22.2V and 16Ah. More information on wiring, connectors and the source code can be provided on request.

A. AI Component

Two AI architectures matching our requirements are YOLO v3 [38] and Faster R-CNN with Inception v2 [39]. Both networks predict objects on different scales to avoid vanishing gradients with more complex models. The approach of each neural network differs in regards to creating bounding boxes but also in fundamentals such as convolution size.

YOLO v3 gains its real-time performance benefits by working with fixed sub-image sizes and bounding boxes [38]. In contrast, Inception V2 focuses on smaller convolution sizes and auxiliary classifiers [39]. To avoid incorrect classification due to over-fitting to a certain object type, we selected Inception V2 as our deep neural network for on-device computing, although YOLO v3 can process more images per second [40]. Following the benchmark results by NVIDIA developers on their Jetson TX2 Nano platform, a double digit amount of frames per second can be achieved when utilizing the Inception V2 network [41].

The Inception V2 with 216×216 pixels in the input layer is smaller compared to the resolution of YOLO v3 with 416×416 pixels. Inception v2 also utilizes 42 layers in comparison to YOLO v3 with 53 layers with convolutions of size 3×3. In addition, Inception V2 computes faster compared to others networks such as VGG [39] with convolutions of the same size [27]. Smaller convolutions packed in groups are also advantageous for low memory GPUs such as the NVIDIA Jetson TX2 Nano, which we decided to use. In case of a memory shortage, currently unused convolutions can be swapped out of memory.

We chose a combination of Faster R-CNN with Inception V2, which provides us with robust results in recognition, good training capabilities, acceptable image resolution and low computation time [40]. The Faster R-CNN approach for region selection [42] enables us to switch the internal neural network architecture if needed.

IV. HARDWARE AND SOFTWARE ARCHITECTURE

Figure 2(a) shows the hardware architecture. The Pixhawk 2.1 Cube flight controller is connected to a Here 2 GNSS for better accuracy and access to the satellite navigation networks GPS, Galileo, GLOSNASS and BeiDou. Additionally, we connected it to our fallback remote control via the FrSky X8R. This is required by regulatory standards in Germany and allows an operator to take control of the drone in an emergency. In normal operation mode however, the drone flight is controlled on a higher level by the NVIDIA Jetson computing unit using the Robot Operating System (ROS) framework [43]. It controls the Pixhawk by transmitting target coordinates. In this way, it also manages object avoidance. To be able to recognize objects, it is connected to a CAA L-8 system that transmits its sensor data via UART and to an additional LidarLite v3 Sensor that measures distance to the ground.

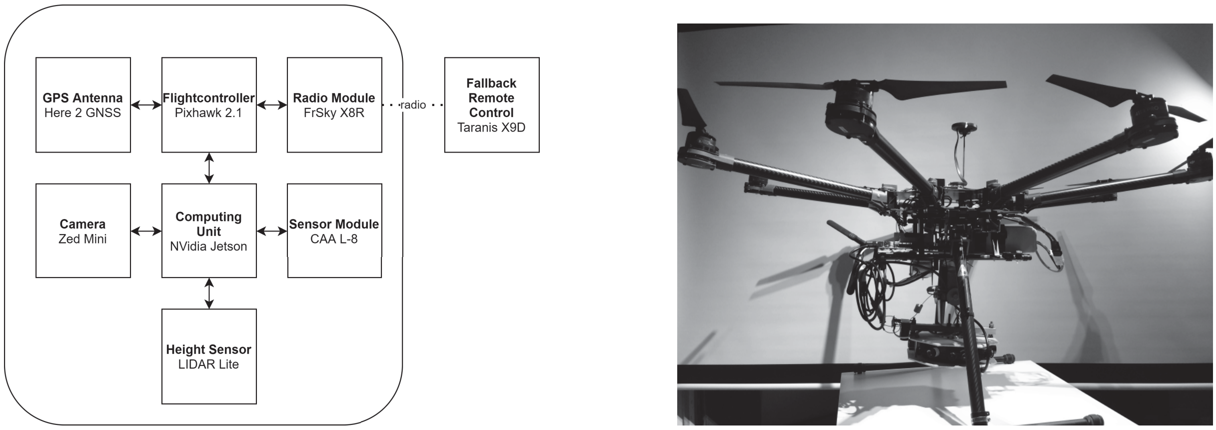
The imagery taken by the Zed Mini is directly forwarded to the NVIDIA Jetson that analyzes these pictures on-device and in real-time. The results and images are saved on an internal storage for potential improvement of the deep learning model and the data base. The distance to the tree tops is reported by the zed mini and the height sensor. Together with topological maps from the Bayerische Vermessungsverwaltung [44], tree height can be calculated.

In addition, we are able to transmit the results directly via Wi-Fi to the user. With this data, a forester will be able to inspect trees of interest.

V. AUTONOMOUS DRONE MOVEMENT

For optimal results, the drone should fly autonomously. This reduces manual labor and helps to keep a steady distance to the trees, which increases accuracy of the collected imagery. In order to be able to fly autonomously, the drone includes a flight plan builder and a simple object avoidance system.

The software QGroundControl [45] generates a flight plan, which then can be transferred to the drone itself. It receives a set of GPS coordinates which it then uses to generate its flight path. The drone smooths the movement of the drone internally, which reduces motion blur on the images. As flying



(a) Schematic of the hardware architecture

(b) M600 drone with LIDAR, Camera, Pixhawk, and Jetson GPU

Fig. 2. Overview of the hardware used in the presented experiments

purely on the data provided by QGroundControl would rely on information that can be out-dated, an additional object avoidance system is required. Also, the possibility of moving obstacles such as cars, humans and animals strongly implies adding an object avoidance system.

The object avoidance system we implemented relies on LIDAR and ultrasonic sensors on a horizontal plane and a simple LIDAR height sensor. When an obstacle is detected, the mission is paused and the height is adjusted so that the drone can raise its altitude and continue its mission.

VI. Tree Detection and Localization

The task of the AI is to detect and map every tree in a forest for further data aggregation and monitoring. With the marked trees and their GPS coordinates, a forestry office is able to monitor and plan accurately.

A. Data and Labeling

Our initial data set consist of open source images taken from drones or planes with specific angle and distance found on image platforms such as Flickr. We are not able to train our neural network with satellite image because of different altitude and resolution. The suitable analogy would be the training of an AI in a simulated environment and improve the weights over time with real data, resulting in a failure due to unknown factors and differences in the image pixel manifolds [46].

We labeled an average of 20 trees on each of 50 images, resulting in approximately 1000 labeled trees. The low number of trees per image is due to the altitude and the uncertainty in labeling. We decided to label only the trees that can be clearly identified to avoid creating errors in the training and evaluation data for our AI.

Other approaches, such as simulated environments and data augmentation, were not used because of high complexity and unknown factors in simulating natural growth processes of trees, for example with fractals [47]. Trees usually follow a specific growth schema considering available light, distance to other trees, adaption of the tree to its environment, time of year, weather, and others [48]. Therefore, simulating this complex data was out of the scope for this project. The high variability of forest image data leads to the rejection of additional data augmentation techniques, including changing color of the image.

B. Segmentation and Bounding Boxes

Several challenges arise for image segmentation in case of forests or trees. Trees of the same species in images often have no clear outline marked by a color change. Additionally, differentiation between background and foreground is hard to achieve with images from high altitude. Saha applied a state-of-the-art neural network to the straightforward problem of road detection with limited results [49].

To avoid this, we choose an object detection method with boxes that meet an adjustable confidence threshold. Thus, the risk of not identifying blending trees as one tree is lower, as a neural network also learns the shape of the boxes and thus the shape of the tree and its limits. In Addition, with the bounding boxes from the deep neural network the tree diameters can be approximated.

C. Architecture

The researchers of TreeSegNet propose a neural network to localize trees in aerial images [26]. They identified the challenge of easily confused classes in tree classification and detection. Their goal was to differentiate trees from other classes in aerial images. However, we and are not interested in other classes and already know that there are trees, as the user wants to map his forest.

The architectures YOLOv3 and Faster R-CNN with Inception v2 were considered as they meet our requirements. The Custom Object Detector is based on the Faster R-CNN with Inception v2 provided by Tensorflow implementation with a pre-trained model to classify objects into the 90 different categories of the MSCOCO data set as presented by [50]. This model was selected due to its high accuracy and reliability in detecting and marking objects with a bounding box [40]. The MSCOCO data set contains common objects such as bottles, cars, and cats. Around 10 000 out of 886 000 instances of all segmented objects are dedicated to potted plants [51]. This provides opportunity for transfer learning.

An example of our initial data is shown in Figure 3. Our data set is not large and diverse enough to train to train a custom deep neuronal network from scratch. Thus, a transfer learning approach was chosen to decrease training time and increase robustness of the network against over- and under-fitting.

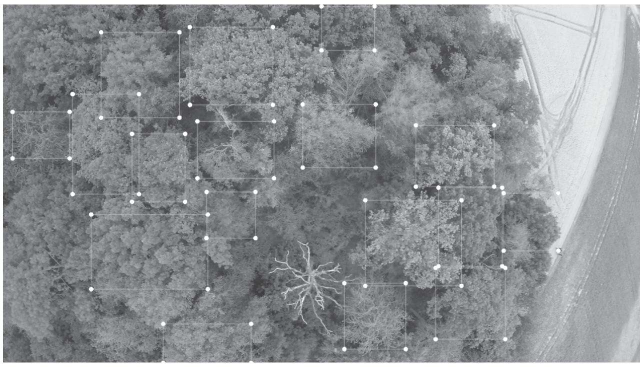


Fig. 3. Example image of multiple trees labeled with LabelIMG

With transfer learning, we take advantage of the ability of detecting different objects from the pre-trained and robust model. We modified and retrained it to detect trees, since our interest in the local cat population was limited. Since we only classify two categories, tree or no tree, our data proved sufficient for acceptable results.

VII. Evaluation

The presented platform, consisting of the flying platform and the AI software, will be evaluated in the following sections. The AI evaluation is split into the training phase and a brief discussion.

A. Flying Platform

The cost of the system sums up to approximately 1 500. This is still low compared to the costs of aerial and satellite based data. As DJI stopped producing the S1000 frame, a frame with similar capabilities would be required to extend or reproduce this project.

Our platform was able to fly over forests, successfully avoided contact with vegetation and took images that can be used for tree detection. Combining distance measurements to tree tops from the Zed-Mini with topological ground maps from the Bayerische Vermessungsverwaltung [44] allowed us to calculate height information for individual trees. Flight time was around 20 minutes, so for commercial application, we recommend a battery casing that can be swapped quickly.

B. Training and Evaluation of Faster R-CNN with Inception V2

Our limited, but sufficient amount of 50 images with around 20 object instances per image are trained for 10 000 steps, with evaluation every 500 steps in between. The neural network can train to detect trees from around 1000 labeled object instances. The high amount of training steps compared to the available data is necessary for the transfer learning approach. We targeted small weight changes over longer periods of time to diminish probability of catastrophic forgetting [52].

Following the 80/20 rule for splitting the data in training and evaluation sets [53], 10 images are taken for evaluation of the Faster R-CNN Inception V2 network. The approach of splitting a data set does not prevent over-fitting of a network, but lessens the probability by continuous evaluation during the training. Approximately every 500 steps, the network loads a snapshot of the currently trained networks and evaluates it using the evaluation data set. Training and evaluation are done on two GPUs, which enables us to execute it simultaneously but also leads to variations in the intervals of the evaluation.

We enabled dropout with a rate of 0.2 to increase the generalization of the network. We also enabled the dropout without retraining the network with the entire COCO data set. This decreased the learning capabilities of the network, but decreased over-fitting.

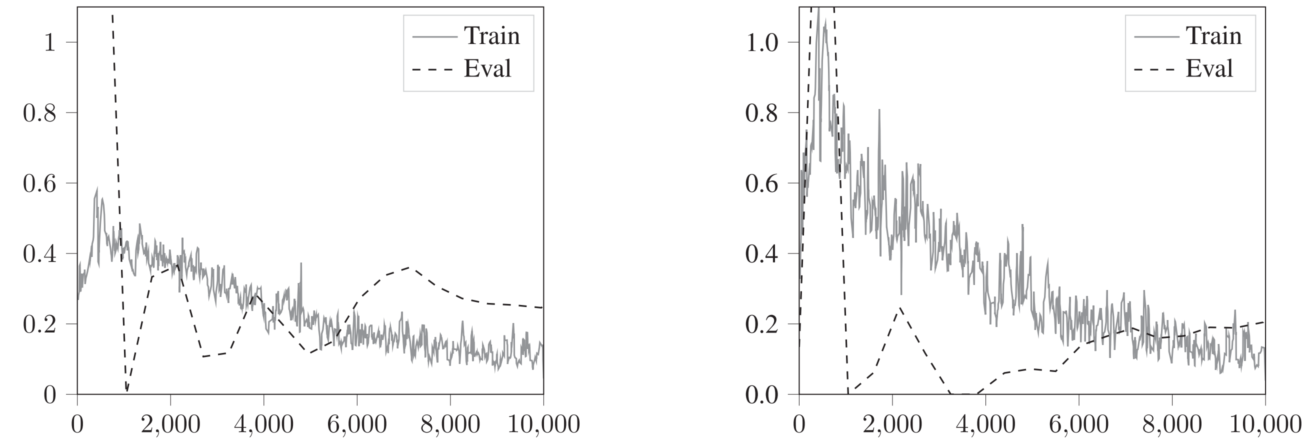
Training parameters are set to a batch size of two images, the use of a momentum optimizer and a slightly increasing warm-up learning rate for the first 2 500 steps. The remaining 7 500 steps were trained with a cosine decaying learning rate at the base of 0.16.

In Figure 4 the results of our training are presented. The localization and classification loss decrease over time. The loss for the classification of trees ended at 0.22 and the loss of the localization at 0.19. We smoothed the values for a better visibility the training loss with a moving average of 5 values. As seen in Figure 4, the evaluation values fluctuates at the low end during the training. We are concerned about the increase of the evaluation value towards the end of the training. We suspect, this could be due to the limited number of object instances contained in the training data and the comparatively high number of training steps. In the future, we will utilize this drone platform to significantly increase the number of images and retrain the network again which we expect to result in a lower evaluation loss. Currently, the evaluation loss for the classification and localization are at 0.25 and 0.21, respectively. We accepted trees detected with a confidence of at least 75%.

C. Discussion of AI

A pre-trained neural net with 90 original classes was transferred to a model with two classes. This may lead to over- or under-fitting, so we have successfully made some adjustments to limit this. Our model can recognize tree and non-tree objects, which needs to be extended in further works to differentiate tree species and tree related object such as bushes. Other deep learning models such as Detectron2 with object segmentation in its focus would also provide similar data but are expected to also merge multiple trees into one.

Our transfer trained detector's loss values vary between 0.15 and 0.25. While this might be improved by adding more and



(a) Localization loss of our detector

(b) Classification loss of our detector

Fig. 4. Localization and classification loss of Faster R-CNN Inception v2 after 10000 training steps with an evaluation every 500 steps. The training graph has been smoothed with a moving average over 5 values

diverse training data, this work shows that the trained detector is capable of classifying and localizing trees.

VIII. Conclusion

Our drone platform achieves a stable flight with all necessary equipment and sensors above the targeted area without colliding with any obstacles. The user is able to plan a flight path, which the drone will then follow, hovering above the trees and takes images of the canopy. The on-board computer processes these images using a deep learning approach to identify and localize individual trees. The inference speed and accuracy of the localized trees computed on the Jetson TX2 Nano meet our requirements as discussed in Section II-B.

Using the point cloud and a digital terrain model, the tree heights are calculated. Using the bounding box sizes from the deep neural network, the tree diameters are approximated. This data is then stored with the GPS position of the tree and prepared to be extracted after landing.

IX. Future Work

This paper presents a starting point for discussions on how to set up a drone for gathering data about trees. We encourage everyone to employ a similar architecture to create an extensive data set of trees, landmarks or other features of interest. We are always happy to share data sets, drone architectures and discuss possible improvements to the presented architecture.

If more airtime is required or a larger area has to be surveyed, the drone concept could be adjusted. A tilt-wing drone design could provide a longer flight time while still not needing a starting and landing strip. However, the flight plan would become more difficult to construct, as changing direction needs either more space or a change of flight mode from horizontal to VTOL flight mode and back. Also, terrain with cliffs might impose a challenge for the object avoidance system and might require a higher minimum distance to the forest. This change could lead to higher requirements for the camera resolution as the distance to the trees increases.

As we managed to propose a system that allows creating a database of trees and some information, the information could still be extended by more data or sensors. Tree health information can be gathered by multi-spectral sensory.

Also, differentiation of tree species based on the collected data might be added. With a growing data base, supervised machine learning could be applied to implement this feature.

For repeated flights over the same area, the flight plan could also be further optimized by using the collected height data to improve path finding and flight duration.

With this platform, we plan to collect forest images in the desired amount of at least 1000 images with at least 10000 object instances per tree species. With that data, we expect to improve our AI significantly and add tree species detection.

Acknowledgment

We want to thank Alexander Koch, Paul Pflugradt, Maximilian Wagner and Lars Wiesner for their work on this project.

References

[1] C. Schmidt. The forests in germany. [Online]. Available: https://www.bmel.de/SharedDocs/Downloads/EN/Publications/ForestsInGermany-BWI.pdf

[2] E. Setyawan. Satellite imagery: Resolution vs. accuracy. [Online]. Available: https://www.intermap.com/blog/satellite-imagery-resolution-vs.-accuracy

[3] NASA. Landsat 7. [Online]. Available: https://landsat.gsfc.nasa.gov/landsat-7

[4] C. J. Fettig, K. D. Klepzig, R. F. Billings, A. S. Munson, T. E. Nebeker, J. F. Negrn, and J. T. Nowak, "The effectiveness of vegetation management practices for prevention and control of bark beetle infestations in coniferous forests of the western and southern united states," *Forest Ecology and Management*, vol. 238, no. 1, pp. 24–53, 2007. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378112706010310

[5] B. L. fr Wald und Forstwirtschaft. Borkenkferinfoportal. [Online]. Available: https://www.lwf.bayern.de/waldschutz/monitoring/065609/index.php

[6] M. Simard, N. Pinto, J. Fisher, and A. Baccini, "Mapping forest canopy height globally with spaceborne lidar," *Journal of Geophysical Research (Biogeosciences)*, vol. 116, pp. 4021–, 12 2011.

[7] A. X. Guo, N. Coops, S. Gergel, C. Bater, S. Nielsen, J. Stadt, and M. Drever, "Integrating airborne lidar and satellite imagery to model habitat connectivity dynamics for spatial conservation prioritization," *Landscape Ecology*, vol. 33, 03 2018.

[8] J. Zhang, S. Nielsen, L. Mao, S. Chen, and J.-C. Svenning, "Regional and historical factors supplement current climate in shaping global forest canopy height," *Journal of Ecology*, vol. 104, pp. 469–478, 03 2016.

[9] R. Nelson, "How did we get here? an early history of forestry lidar 1," *Canadian Journal of Remote Sensing*, vol. 39, pp. S6–S17, 06 2014.

[10] K. Anderson and K. Gaston, "Lightweight unmanned aerial vehicles will revolutionize spatial ecology," *Frontiers in Ecology and the Environment*, vol. 11, pp. 138–146, 03 2013.

[11] T. Sankey, J. Donager, J. McVay, and J. B. Sankey, "Uav lidar and hyperspectral fusion for forest monitoring in the southwestern usa," *Remote Sensing of Environment*, vol. 195, pp. 30 – 43, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0034425717301578

[12] D. Almeida, E. Broadbent, A. Zambrano, B. Wilkinson, M. Ferreira, R. Chazdon, P. Meli, E. Gorgens, C. Silva, S. Stark, R. Valbuena, D. Papa, and P. Brancalion, "Monitoring the structure of forest restoration plantations with a drone-lidar system," *International Journal of Applied Earth Observation and Geoinformation*, vol. 79, pp. 192 – 198, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0303243418311954

[13] J. Zhang, J. Hu, J. Lian, Z. Fan, X. Ouyang, and W. Ye, "Seeing the forest from drones: Testing the potential of lightweight drones as a tool for long-term forest monitoring," *Biological Conservation*, vol. 198, pp. 60 – 69, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0006320716301100

[14] F. Morsdorf, E. Meier, B. Ktz, K. I. Itten, M. Dobbertin, and B. Allgwer, "Lidar-based geometric reconstruction of boreal type forest stands at single tree level for forest and wildland fire management," *Remote Sensing of Environment*, vol. 92, no. 3, pp. 353–362, 2004, forest Fire Prevention and Assessment. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0034425704001464

[15] K. Yue, L. Yang, R. Li, W. Hu, F. Zhang, and W. Li, "Treesegnet: Adaptive tree cnns for subdecimeter aerial image segmentation," 2018.

[16] M. Kampen, S. Lederbauer, J.-P. Mund, and M. Immitzer, "Uav-based multispectral data for tree species classification and tree vitality analysis," in *Dreiländertagung der DGPF, der OVG und der SGPF in Wien, sterreich Publikationen der DGPF, Band 28*, 01 2019.

[17] Z. Zhao, P. Zheng, S. Xu, and X. Wu, "Object detection with deep learning: A review," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 11, pp. 3212–3232, 2019.

[18] Z. Liu and S. Wang, "Broken corn detection based on an adjusted yolo with focal loss," *IEEE Access*, vol. 7, pp. 68 281–68 289, 2019.

[19] Y. L. Kuo and S. H. Lin, "Applications of deep learning to road sign detection in dvr images," in *2019 IEEE International Symposium on Measurement and Control in Robotics (ISMCR)*, 2019, pp. A2–1–1–A2–1–6.

[20] D. Mery, I. Mackenney, and E. Villalobos, "Student attendance system in crowded classrooms using a smartphone camera," in *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2019, pp. 857–866.

[21] B. fr Kartographie und Geodsie. (2021) Digitale topographische karte dtk25 (digital topographic map. [Online]. Available: http://sg.geodatenzentrum.de/web_bkg_webmap/applications/info/dtk25.html

[22] M. Pak and S. Kim, "A review of deep learning in image recognition," in *2017 4th International Conference on Computer Applications and Information Processing Technology (CAIPT)*, 2017, pp. 1–3.

[23] P. Grover. Evolution of object detection and localization algorithms. [Online]. Available: https://towardsdatascience.com/evolution-of-object-detection-and-localization-algorithms-e241021d8bad

[24] N. O. Mahony, S. Campbell, A. Carvalho, S. Harapanahalli, G. A. Velasco-Hernández, L. Krpalkova, D. Riordan, and J. Walsh, "Deep learning vs. traditional computer vision," *CoRR*, vol. abs/1910.13796, 2019. [Online]. Available: http://arxiv.org/abs/1910.13796

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015. [Online]. Available: http://arxiv.org/abs/1512.03385

[26] K. Yue, L. Yang, R. Li, W. Hu, F. Zhang, and W. Li, "Treesegnet: Automatically constructed tree cnns for subdecimeter aerial image segmentation," *CoRR*, vol. abs/1804.10879, 2018. [Online]. Available: http://arxiv.org/abs/1804.10879

[27] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015. [Online]. Available: http://arxiv.org/abs/1409.1556

[28] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional

neural networks," in *Advances in Neural Information Processing Systems*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds., vol. 25. Curran Associates, Inc., 2012. [Online]. Available: https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf

[29] DJI. S1000+ frame with rotors. [Online]. Available: https://www.dji.com/de/spreading-wings-s1000-plus

[30] ——. (2020) M600 pro. [Online]. Available: https://www.dji.com/de/matrice600-pro

[31] ——. (2020) A3 pro. [Online]. Available: https://www.dji.com/en/a3/info

[32] Pixhawk. (2020) Pixhawk 2. [Online]. Available: https://docs.px4.io/v1.9.0/en/flight_controller/pixhawk-2.html

[33] ProfiCNC. Here 2 specifvcation. [Online]. Available: http://www.proficnc.com/all-products/152-here2.html

[34] EmQopter. Lidar caa l-8. [Online]. Available: https://www.emqopter.de/de/caa_lidar.php

[35] Garmin. (2021) Lidar-lite v3. [Online]. Available: https://buy.garmin.com/de-DE/DE/p/557294

[36] FrSky. Taranis x9d plus. [Online]. Available: https://www.frsky-rc.com/product/taranis-x9d-plus-2/

[37] ——. Taranis x8r. [Online]. Available: https://www.frsky-rc.com/product/x8r/

[38] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *CoRR*, vol. abs/1804.02767, 2018. [Online]. Available: http://arxiv.org/abs/1804.02767

[39] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," *CoRR*, vol. abs/1512.00567, 2015. [Online]. Available: http://arxiv.org/abs/1512.00567

[40] J. Hui. (2018) Object detection: speed and accuracy comparison (faster r-cnn, r-fcn, ssd, fpn, retinanet and yolov3. [Online]. Available: https://jonathan-hui.medium.com/object-detection-speed-and-accuracy-comparison-faster-r-cnn-r-fcn-ssd-and-yolo-5425656ae359

[41] N. Corp. Jetson nano: Deep learning inference benchmarks. [Online]. Available: https://developer.nvidia.com/embedded/jetson-nano-dl-inference-benchmarks

[47] N. Lorimer, R. Haight, R. Leary, and M. North Central Forest Experiment Station (Saint Paul, *The Fractal Forest: Fractal Geometry and Applications in Forest Science*, ser. General technical report NC. North Central Forest Experiment Station, Forest Service, U.S. Department of Agriculture, 1994. [Online]. Available: https://books.google.de/books?id=aZp0kj1IBwMC

[42] R. B. Girshick, "Fast R-CNN," *CoRR*, vol. abs/1504.08083, 2015. [Online]. Available: http://arxiv.org/abs/1504.08083

[43] I. Open Source Robotics Foundation. (2021) Robot operating system. [Online]. Available: https://www.ros.org/

[44] B. Vermessungsverwaltung. Digitales gelndemodell. [Online]. Available: https://geodatenonline.bayern.de/geodatenonline/seiten/dgm_info

[45] D. Project. (2021) Qgroundcontrol. [Online]. Available: http://qgroundcontrol.com

[46] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. The MIT Press, 2016.

[48] A. Weiskittel, D. Hann, J. Kershaw, and J. Vanclay, *Forest Growth and Yield Modeling*. Wiley, 2011. [Online]. Available: https://books.google.de/books?id=T12Y8gxNlKkC

[49] A. Saha. U-net-satellite; segmentation based road detection framework using keras. [Online]. Available: https://github.com/ArkaJU/U-Net-Satellite

[50] G. Yadam. Object detection using tensorflow and coco pre-trained models. [Online]. Available: https://medium.com/object-detection-using-tensorflow-and-coco-pre/

[51] T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: common objects in context," *CoRR*, vol. abs/1405.0312, 2014. [Online]. Available: http://arxiv.org/abs/1405.0312

[52] J. Kirkpatrick, R. Pascanu, N. C. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, D. Hassabis, C. Clopath, D. Kumaran, and R. Hadsell, "Overcoming catastrophic forgetting in neural networks," 2016. [Online]. Available: http://arxiv.org/abs/1612.00796

[53] I. Guyon, "A scaling law for the validation-set training-set size ratio," 1997.