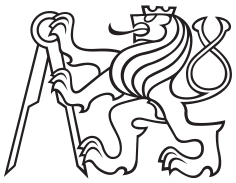
Master Thesis



Czech
Technical
University
in Prague

F3

Faculty of Electrical Engineering
Department of Cybernetics

Detection and Tracking of Objects on Water Surface

Bc. Adam Ukleh

Supervisor: doc. Ing. Martin Saska, Dr. rer. nat.
May 2022

Acknowledgements

I would like to thank my thesis supervisor doc. Ing. Martin Saska, Dr. rer. nat. for his support, guidance, patience and time invested into this work. Furthermore I would like to thank Ing. Pavel Stoudek for his assistance during the collection of the datasets and Ing. Matouš Vrba for his advices during this work. Another thank belongs to RNDr. Petr Štěpán, Ph.D. for reviewing my thesis and his advices.

My big thank you is dedicated to my family for their support during my whole studies.

Declaration

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 obeserving the ethical priciples in the preparation of university theses.

In Prague, 20. May 2022

Abstract

The main goal of this diploma thesis was to design and implement a system that can detect, classify and track the floating debris on the water surface. Methods based on deep learning were proposed and implemented for the detection and classification of floating debris. A fast and computationally not demanding method, which was able to track multiple objects on the water surface, was proposed and implemented. Dataset was collected for the training and evaluation of the deep learning models. The dataset was also used to create videos for the evaluation of the tracking method. At the end of the thesis, we present and discuss the experiments and results. We proposed a detection model, suitable for implementation with the multi-object tracking method on the real hardware of the UAV.

Keywords: unmanned aerial vehicles (UAV), floating debris, object detection, multi-object tracking

Supervisor: doc. Ing. Martin Saska, Dr. rer. nat.

Abstrakt

Hlavným cieľom tejto diplomovej práce bolo navrhnúť a implementovať systém, ktorý dokáže detekovať, klasifikovať a sledovať plávajúci odpad na vodnej hladine. Na detekciu a klasifikáciu plávajúceho odpadu boli navrhnuté a implementované viaceré metódy, ktorých základ bol v hlbokom učení. Navrhnutá a implementovaná bola rýchla a výpočetne nenáročná metóda, ktorá zvláda sledovať viacero objektov na vodnej hladine. Pre natrénovanie a vyhodnotenie modelov so základom v hlbokom učení bol nazbieraný dataset. Dataset sa využil aj na vytvorenie videí, ktoré boli neskôr použité k vyhodnoteniu sledovacej metódy. V závere našej práce prinášame prehľad o prevedených experimentoch a diskusiu o ich výsledkoch, na základe ktorých sme navrhli detekčný model, ktorý je vhodný na použitie so sledovacou metódou pri implementácii na reálny dron.

Klíčová slova: bezpilotné lietadlá, odpad na vode, detekcia objektov, sledovanie objektov

Contents

Figures

Tables

Chapter 1

Introduction

One of the most important resources for living on our planet is water. Millions of tons of trash are thrown into the oceans, seas and rivers every year. This results in polluted waters and finding animals bodies filled with debris. In particular, about 8 millions items of debris are thrown to water environment every day [1]. According to United Nations world water development report, around 3.5 million people die from water infections [2]. Removing and monitoring marine debris has been one of the biggest environmental challenges in the past years. Most of the marine debris monitoring and collecting is performed via boat surveys, which is time and cost demanding and also the human error can occur.

Research and development in computer science and robotic fields have risen in recent years. This gave us the opportunity to automate most of the tasks also in monitoring and removing marine debris. It offers to perform surveys in a maritime environment with less search time, lower cost of expeditions, increased accuracy and human error-free results. Autonomous robotic vessel platforms for detecting and collecting debris on small water bodies were deployed [3]. These platforms can autonomously navigate in a riverine environment, detect and collect debris.

Nowadays unmanned aerial vehicles (UAVs), also called drones, gained popularity. They are used to tackle also marine environment challenges with relatively low price and easy deployment. UAVs give us a flexible platform to carry out cameras and sensors for monitoring marine debris [4]. We can detect and classify small objects, which can be unseen by the human eye with using deep learning approaches from images collected by UAVs [5].

1.1 Motivation and problem definition

Our motivation is to contribute to tackle the global problem with floating marine debris using UAVs. This thesis is a part of a project to develop UAVs capable of removing floating marine debris from the water. The main idea is that UAVs will fly above water and search for floating marine debris. Detected debris will be classified into the most common marine debris categories, which can be found in the marine environment, such as plastic bottles, plastic bags, food packaging etc. [6]. After the detection of floating debris, UAV will start tracking the detected and classified objects. Information about the position of tracked floating debris will be used for grasping it from water.

Our part of the overall solution is focusing on proposing and implementing the detection-tracking system, which can run in real-time. In order to achieve our goal, we need to choose the right payload, which consists of cameras and sensors. The reason is to manage the UAV to be able to sense and see surrounding floating debris.

Our task can be divided into two subtasks. The first subtask will tackle the detection and classification part. For this subtask, we will propose deep learning methods, which will be able to successfully detect and classify floating debris. To use modern deep learning methods, we need to create a dataset for training the models. For a dataset collection we need to choose a camera, which can be able to take pictures of floating debris with a frame rate of 30 frames per second (FPS) and higher. The camera will be mounted perpendicular to the water surface.
Another useful information available during dataset collection is the flight altitude. This information will later be important to determine from which altitute our models can successfully detect and classify the floating debris.
We chose to classify three most common marine debris categories, which are plastic menu boxes, plastic bags and plastic bottles. Collection of dataset will be on the water surface without obstacles, that can overlap floating debris.

The second subtask will tackle the visual multi-object tracking of detected floating debris on the water surface. We will propose a method that will be able to track multiple objects and will not add the computational load to the hardware of the UAV to maintain real-time performance, since the deep learning methods that will be used for detection and classification subtask are computational demanding.

1.2 Thesis outline

In the beggining we explore related work to our problem.
Next we present our proposed methods for solving each of the subtasks. Then we describe the process of creating the dataset with the examples of collected images, which will be then used for experimentation with proposed methods. After the implementation of the proposed methods and performing experiments on our datasets, we will present the achieved results and comparisons. Finally we will conclude and discuss the results and possible limitations of the proposed system.

Chapter 2

Related work

In this chapter, we present related work to two main subtasks of our detection-tracking system, found in the available literature. First part is dedicated to object detection and classification of floating debris on the water surface. We explore which methods and cameras are being used nowadays for tackling this problem.
The second part describes methods for visual object tracking.

2.1 Object detection and classification on the water surface

Object detection of floating debris on the water surface is a challenging task. Floating objects can be partially submerged. Some objects can fully sink under the water and later resurface due to the waves. Plastic bags can often change their shape that vary in time, as they float on the water surface. The water surface can also reflect the sun and the sky. Large waves can also produce foam, which can add unwanted noise and confuse the object detector [7].

Thanks to the recent development in machine learning, especially in subfield of deep learning. These methods started to get more attention in the last years. Most of the object detection and classification tasks are build on deep learning methods, such as convolutional neural networks.
For the static monitoring of floating debris in the city canals cameras mounted

on the bridge construction are used. In [7] state of the art deep learning models are used and compared with adjusted attention layer for focusing on smaller objects. Other example is shown in [8], where they explore an object detection based on convolutional neural networks and generalization of using trained model in other locations with the same environment. Focus is also given on counting the detected objects and comparation of the model with the human counting. According to [9] deep learning methods are also used in detection and counting of floating debris in a river.
Exhaustive study was gathered on detection and classification in riverine environment [10]. Study explored the use of deep learning model YOLOv4 for object detection of 5 classes. These classes were composed of plastic bottles, bags, styrofoam, aluminium cans and plastic containers. During the training, the image augmentations are applied to original dataset for its expansion. Results show different metrics for evaluation of applied deep learning model on detection of floating debris.
In maritime environment object classification was explored using convolutional neural network named VGG16. High accuracy of classification was observed on 3 classes, which were plastic bottles, straws and buckets [11].

For the monitoring of objects on a large water surfaces, Zhang et al. [12] used unmanned surface vehicle (USV). USVs have application mainly in civil and military missions. These vehicles need to be aware of their surroundings for completing their tasks without collisons. USV is equiped with camera mounted on its construction. Visual-based approach is used with deep learning method. Experiments on changing the model architecture proved more mined hidden information and better feature extraction, which resulted in overall better network precision. Real-time detection performance is shown also with high detection results. K-Means clustering algorithm is used to select right anchor boxes, which deep learning model uses for bounding the objects.

Autonomous robots, which are moving on a water surface, are also engaged in the issue of collecting floating debris from water surfaces. In [13] they present small capture robot, which moves on water surface and collects floating water debris. The main focus was on a real-time object detection and classification. A binocular camera is used for capturing the surounding environment. In this case, state of the art deep learning model YOLOv3 is used with modifications. Modifications are focused on making the model faster. Instead of model having three scale detection, two scale detection is presented. This modification makes model less computationally demanding. Results are showing 54 FPS for detection of object with high precision for three main classes, which were plastic bottles, bags and styrofoam.
Another autonomous robot was deployed and tested in real environment. Pi Cam is used to capture images. Object detection is performed by deep learning model which has lighter architecture then original model, which leads to less computational demand on resources [3]. The main target of detection

were plastic bottles.

UAVs are attracting the attention of researchers in recent years, thanks to their broad utilization. In maritime environment, UAVs are mostly used as the assistance in search and rescue (SAR) missons. In these missions, the time and accuracy are very crucial. In [14] they present autonomous UAV, which is used in SAR missions and is equiped with thermal camera. Proposed deep learning method, called Faster R-CNN, can detect people from different altitudes in images taken by the thermal camera.
UAVs are also equiped with RGB cameras for detecting small targets. Comparison of the detection of small targets between human and SSD deep learning model was examined [5]. Deep learning model can detect targets faster then human eye by 17 seconds. Strategy behind sucessfull detection was in splitting the high resolution video with 4K quality to small images and enhancing the contrast of each image for better detection. The model was detecting targets in every small image. After performing detection in small images, these images were formed into original high resolution image with detected targets.
In a real-time search and rescue operations, target detection speed and accuracy must be balanced. Especially in UAV this balance is bound to flight altitude. More area can be searched and detected with the higher altitude, on the other hand more precisely can objects be detected with smaller altitude [15].

UAVs are starting to being used in monitoring of floating marine debris and water surface objects. Plastic bottles, plastic bags, drifting wood and plastic trays were detected by state of the art deep learning model deployed on UAV hardware in real-time [4]. Different altitudes for object detection were observed and best altitudes for sufficient object detection were under 30 m. Visible light camera is used for capturing the images. Properties and future of including IR camera is discussed.
Interesting system which consists of three main modules was developed. First module is focused on autonomously changing the battery of UAV, which landed on docking station. Second module has in charge the communication between modules. The last module is performing monitoring of water surface and its main target is to detect floating debris. Object detection is performed in real-time and modified deep learning model is used [16].
Another study by Zhang et al. [17] compared deep learning models for object detection on water surface. New layer for mining more features from input images improved detection of smaller objects and increased the accuracy of YOLOv3 model. Improved model had the highest accuracy, in comparison with other models on custom dataset. Images were taken by RGB camera mounted perpendicular to the water surface. Custom dataset consists of sand dredges, aquatic plants, fishing boats, green algae and reeds. Dataset was also expanded by performing image augmentations as rotation, brightness

augmentation and mirroring.

2.2 Object tracking

2.2.1 Object tracking on the water surface

Complex and fast changing marine environment makes tracking of floating objects on water surface more difficult, than tracking objects on land. Water surface waves can partially submerge tracking objects. Waves can fastly change object's velocity and direction, thus correlation between frames can be disturbed. Sun light can be reflected from floating objects and water, which can cause the overlook of the object by the tracking system.

In the recent years most of the research of object tracking on water surface was targeting tracking single and multiple vessels due to deployment of USVs. USVs need to have information about the position of other vessels in their surrounding for safety navigation. Duarte et al. [18] focused their research on detecting and tracking multiple vessels by so called tracking by detection method. They took a deep learning approach for object detector and tracker. Transfer learning was used to train detection model YOLOV4 and tracking model DeepSORT for a specific task in marine environment. Selected models were able to succesfully overcome challenges in maritime environment such as fog, exposure issues, waves and they were able to detect and track multiple vessels.

In [19] single object tracking method is presented. They used well known single object tracking algorithm called Siamese network with modified subnetwork with multi-RPNs. Advantage of using visual cameras instead of radars and thermal cameras is explored. Visual cameras do not have high energy demand on the system and they are less expensive then other mentioned sensors. They also can provide high details of object, which is in the task of object tracking highly demanded, especially in the use of deep learning based methods. Results of the comparison show, that modified Siamese network outperforms four compared single object trackers (SOT). Limitations of the proposed method are in the need of large dataset and in challenging performance of a tracker in harsh weather conditions.

One of the most problematic situation in object tracking is when other object occlude desired tracked object. Problem of occlusion was tackled by

Chen et al. [20]. The use of kernelized correlation filter (KCF) is proposed for tracking ships and extracting their trajectories. On top of the KCF curve fitting model is implemented for adjusting deviated ship trajectory caused by occlusion. Due to high percentage of collisions between ships and bridges, caused by human error as stated in [21], automatization of processes like detection and tracking on board of the ships are deployed. Multitarget tracking framework based on tracking by detection is developed. Framework can track multiple vessels under occlusion and can reidentify them, which reduces ID switches between tracked objects. Motion of surrounding vessels is predicted with GRU recurrent neural network based on historical motion data of target vessels. Another part of the framework is data association method, which considers short and long term cues. ID switches are being tackled by ship reidentification method which is responsible of deciding, whether the vessel from previous video frame is also present in the current frame. Performed experiments on the framework show real-time performance and robustness. As the results from detector are crucial for tracking by detection paradigm, performance of YOLOv3, SSD300 and Faster R-CNN were studied. Tracking of multiscale ship was studied in [22]. They presented the method composed of target tracking algorithm and re-detection algorithm. Proposed method can run in real-time and can tackle occlusion, blur and deformation of tracked object.

Use of the UAVs in maritime surveillance system for helping vessels to change direction in time, to avoid collisions is shown in [23]. Detection of objects on water surface is done by using deep learning model YOLOv5. Multi object tracking algorithm SORT is used like a filter for removing false positive detections, which enables to lower confidence threshold. Using a combination of visual camera and thermal camera is shown to provide added value to object detection. Thermal camera can deal with sunlight reflection and low illumination. Size of objects and overall input resolution of deep learning detection model have effect on system detection performance. Experiments with different variants of YOLOv5 model are performed. Another study on maritime surveillance, especially detecting and tracking sharks was conducted [24]. Study is focusing on how to remove human in the loop and centralised system architectures by implementing object detection and tracking on UAV hardware. Four object detection models and six tracking algorithms are compared. Making proposed system efficient, structural similarity index is proposed. It measures tracking confidence and frame similarity, thus overall system is balanced between detecting new objects and tracking existing objects.

2.2.2 Object tracking from UAV on land

Most research on object tracking from UAVs is gathered on land. The main reason is that UAVs have nowdays more application on land in border patrol, search and rescue operations after disasters, surveillance and more. In [25] multiple object tracking by leading paradigm is explored. YOLOv3 and RetinaNet are responsible for the creation, update and cancellation of created tracks. RetinaNet can successfully detect objects from higher height than YOLO. DeepSORT tracking algorithm is used. With his ability to extract features of tracked objects by CNN, re-identification of objects is possible. ID switches are also reduced, thanks to the re-identification. Performance of the proposed system is explored on the VisDrone benchmark and compared with three more tracking algorithms, including base model of DeepSORT algorithm. Comparation shows that proposed system has better tracking accuracy than other compared algorithms.

More demand is put on UAVs in terms of execution of actions depending on captured data, which is crucial for autonomous UAVs. Nousi et al. [26] explore implementation of state of the art detection and tracking algorithm on board of the UAV. Single stage detectors are used for their computation efficiency. For achieving much faster tracking, they develop the multithreaded KCF and SiamFClite algorithm. First mentioned is based on well known KCF tracking algorithm. With every obtained frame, three threads are running in parallel and every thread is responsible for different scale factor of the region of interest. Latter one is based on SiamFC tracking algorithm. To make the algorithm perform faster, depth factor $\alpha \in (0, 1]$ is introduced. By this depth factor, number of filters in layers of siamese network are multiplied. This makes network lighter in terms of computational demand. Detection-tracking system is implemented in ROS environment. Speed and accuracy of detection and tracking algorithms are performed on different datasets. Another study of detection and tracking system implemented onboard of the UAV uses JPDA with YOLO algorithm [27]. Proposed algorithm is detection free and it has a low demand on computational resources. The algorithm do not use image information, but with well chosen parameters it can get performance of state of the art tracking algorithms on benchmark dataset. Tracking algorithm can handle detection lose for a short time caused by occlusion or missed detection by object detection model. The bottleneck of proposed algorithm is that UAV motion is unmodeled inside the dynamic model. In comparison with models on top of the chart, 20% drop in performance on MOTA benchmark was observed, but processing speed was higher.

Experiments of following the walking person by UAV in outdoor environment were gathered [28]. New visual tracking algorithm is developed, which improves classic KCF by introducing scale adaptive algorithm. Introduced

improvement can deal with changing size of tracked object. Comparison between three more algorithms shows, that proposed algorithm do not have high computational demand, which is suitable for implementation on UAV hardware. Interesting approach of applying deep neural network for detection and tracking of objects is shown in [29]. The main area of application was city environment, where the objects are moving along limited trajectories and thus can be easily predicted. YOLOv4eff is used as an object detector and doubled LSTM as an object tracker. Doubled LSTM track objects based on their trajectories. Comparison between similar object trackers are showing that proposed method has higher tracking precision and accuracy.
Shen et al. [30] deployed first Siamese tracker on UAV embedded system, due to presented guideline for reducing computational requirements of the network. The main idea behind the guideline is reducing the dimension of feature space by every consecutive layer output in network being smaller than dimension of input. They reduced demand on computational resources of well known feature extractor AlexNet by 59.4% with remained tracker accuracy. Introduced anchor free tracking head also removes number of computations. Proposed Siamese tracker can tackle different sizes of tracked object and partial occlusion. Comparison with state of the art Siamese trackers on benchmark datasets is performed and shows that lightened Siamese network can perform without loss of comparable tracking performance.

Multiple object tracking algorithm, which solves error detections and ID switches caused by missed detections or rapid camera motion is presented [31]. Optical flow network deals with rapid camera motion and it is used for an estimation of motion of two consecutive frames and predicting the position of tracked object. Reduction of false matches is done by cascade matching strategy with use of intersection over union (IoU) and deep features extracted by residual network. Optical flow network is also used as an auxiliary tracker in cases when the tracklet is interrupted. Faster version of the tracking algorithm does not use optical flow for each frame, thus faster tracking can be achieved with comparable accuracy. Speed estimation of moving objects on the ground from UAV is tackled by using tracking by detection method. Tracking algorithm called discriminative correlation filter with CSRT is used with added properties [32]. One of the properties is feature-based image alignment, which is responsible of obtaining appropriate tracked object location. UAV is moving platform and it is important to measure similarity between frames. Similarity is measured by algorithm called structural similarity index measurement, which measures similarity between actual frame and frame where the object was detected. Detection part of the proposed method is performed in the case, where the computed similarity threshold is lower than set value. Performing detection only in these cases leads to lowering computation demand on hardware. The main bottleneck of the algorithm is occlusion. Comparison between static and moving drone shows that difference in speed estimation is only 1%.

Another study of detection and tracking system onboard of the UAV was done in [33]. DeepSORT was used as the tracking algorithm. Importance of training deep metric with large enough dataset is highlighted. Person re-identification dataset consisted of more than one milion images of pedestrians. By using graph neural network object tracking can be tackled as shown in [34]. In the proposed method object detection and association are combined into single model. Centerpoint is used as anchor free object detector and extractor of re-identification features. Association between objects across frames is done in graph network. Extracted features and bounding boxes are passed into graph convolution association. Re-identification features show their importance in maintaning ID of the track under the influence of heavy occlusion. In comparison between tracking algorithms on UAVDT benchmark shows, that tracking accuracy, precision and ID switches of proposed method are improved and algorithm achieved state of the art results.
Exhaustive survey describing recent development in deep learning object detection and tracking was conducted [35]. Nowdays multiple object tracking is based on tracking by detection paradigm. Recent state of the art tracking methods are compared on four benchmark datasets. The use of infrared, multispectral and hyperspectral sensors can provide complementary information and make object detection and tracking performance more precise.

Chapter 3

Proposed solution

3.1 System design

These days, seas and oceans are full of various debris. Autonomous vehicles with vision system able to detect, classify and track the floating debris are needed.
Our main goal is to propose and develop a detection-tracking system, which will be responsible for detection, classification and tracking of the floating debris on water surface. Developed system will be one of the main components of the whole grasping system of the UAV. Since we decided to use nowdays leading paradigm for multi-object tracking in videos, which is tracking by detection, the well performing object detection method is necessary.
Our solution is composed of two parts. First part solves a detection and classification problem. Second part is responsible for multi-object tracking of detected floating debris. In the Figure 3.1 we can see the proposed design of the system.

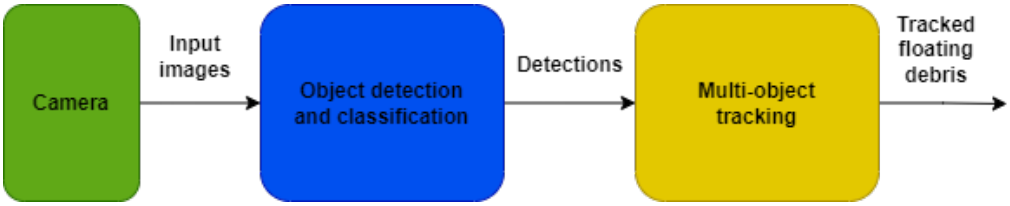


Figure 3.1: Design of the detection-tracking system.

The UAV will be flying above water. Images will be captured by camera mounted on the UAV. The camera will be able to capture images with frame

rate of 30 FPS and higher. Developed system will be using images captured by camera as an input to the object detection and classification part. Our system will be able to classify floating debris to three categories, which are plastic menu boxes, plastic bottles and plastic bags. Modern methods based on deep learning will be used in this part. Proposed methods, which are suitable to fulfill this task, will be described in this chapter.

As we can see from the system design, object tracking algorithm will be placed on top of the object detection and classification part of our system. Tracking algorithm will be responsible for tracking the detected floating debris. Information about the position and class of detected debris in captured data will be later used to navigate the UAV on top of the debris and grasping it from the water.
The method which can be able to track multiple objects will be proposed and described in this chapter. Emphasis will be put on a method that will not add more computational load on hardware besides deep learning model responsible for detection and classification. This is due to future implementation of our detection-tracking system on embedded hardware of the UAV.

3.2 Detection and classification of floating debris

In this section, we will present the description of our solution for tackling the first part of our detection-tracking system, which is object detection and classification. Our system must be capable to determine the position of the floating debris on water surface in image and subsequently identify its class correctly, which is a complex and challenging task. We will describe two methods based on deep learning, which will be used to detect and classify floating debris.

For tackling this challenging task, we can not use the algorithms, which are using features defined by user. Firstly, because we can not cover all the properties of our complex problem with its occuring challenges. Another problem with user defined features is a time exhausting procees of creating and selecting features. Therefore we need to search for methods, that can be teached to extract features automatically from the domain, where they are applied. This leads us to machine learning field.

With the recent development in the machine learning field, especially in a subfield of deep learning, we propose to use deep learning models for this challenging task. Deep learning models can learn to extract the usefull

features automatically. We need to have a lot of data in the input to the deep learning models. Input data are used to teach 'train' the models. The foundation of deep learning models, which are being used nowdays for object detection and classification are convolutional neural networks (CNN's) [36]. CNN's are composed of convolutional layers, pooling layers, non-linearity layers and fully-connected layers. They are named by the linear mathematical operation between matrices called convolution. Function of CNN is shown in the Figure 3.2.

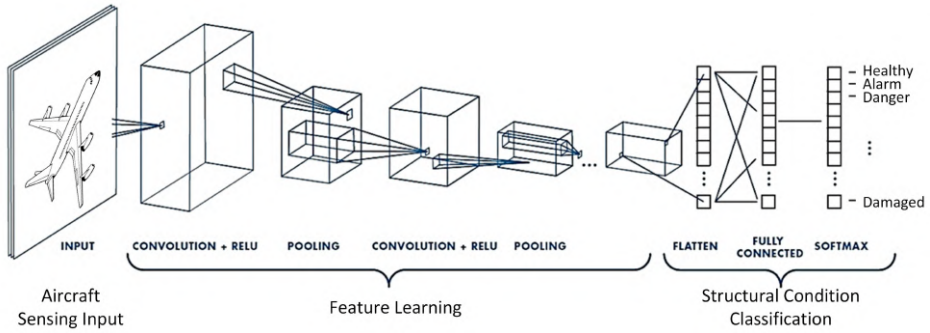


Figure 3.2: Function of CNN [37].

Our solution for tackling a problem of detection and classification of floating debris is based on modern deep learning models. First deep learning model, which we decided to use, is a well known model named YOLO (You only look once)-v3 [38]. Proposed deep learning model is result of a further improvement of YOLO baseline multi-class detection model. YOLOv3 is fully convolutional model, as its predecessors, which means that it consists only from convolutional layers. Name of the model came from the fact, that the deep learning model take the whole image in the input and pass it through the CNN only once. In the output we have predictions of bounding boxes offsets, classes and objectness score. YOLOv3 belongs to category of one stage detectors. One stage detectors are faster than two stage detectors. Two stage detectors are using two networks for obtaining the detections. One network is responsible for region proposal, which determine the position of object and the other network is responsible for the detection.

YOLOv3 model uses a network called Darknet 53 for the feature extractor, which contains 53 convolutional layers. Feature extractor consists of 1x1 and 3x3 convolutions, which are used with skip connections, which are significant for residual networks. Darknet53 is faster than ResNets and the performance stays comparable. On the top of the feature extraction network, more convolutional layers are added. YOLOv3 is detecting objects on three different scales. First detection scale is on convolutional layer number 82, the second is on the layer 94 and the last detection scale is on 106. Each output of the detection scales is divided into N x N grid cells. Three anchor boxes,

also called priors, with different sizes are assigned to each detection scale. Predictions of bounding box offsets, objectness score and classes for every assigned prior are predicted for each cell. Objectness score expresses how well the bounding box overlap the ground truth object. Predicted bounding boxes offsets are offsets to priors. Architecture of YOLOv3 can be seen in Figure 3.3.

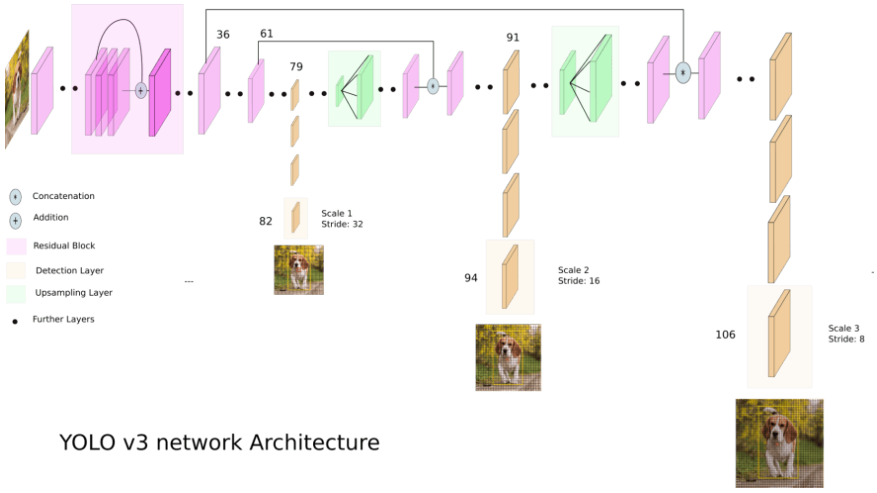


Figure 3.3: Architecture of YOLOv3 model [39].

Four bounding box offsets are predicted for each of the anchor box. These offsets are t_x, t_y, t_w, t_h. Using offsets instead of absolute values can help eliminating unstable gradients. The equations 3.1 show computation of final bouding box position, where c_x and c_y are offsets of the cell from left corner of the image and anchor box has width and height p_w, p_h.

$$\begin{aligned} b_x &= \sigma(t_x) + c_x \\ b_y &= \sigma(t_y) + c_y \\ b_w &= p_w e^{t_w} \\ b_h &= p_h e^{t_h} \end{aligned} \tag{3.1}$$

Sum of squared error loss is used to measure accuracy of predicted bounding box coordinates. Ground truth value can be computed from the inversion of equations 3.1.

Classes, that bounding box may contain, are predicted by using independent logistic classifiers and binary cross entropy is used for the training.
In YOLOv3 only one prior is assigned to each ground truth object. Objectness score of the bounding box is computed by logistic regression instead of softmax, which was used in older versions of the model. Value of the objectness score should be 1, when prior overlaps ground truth object more than any other prior. Overlap between prior and ground truth object is computed by intersection over union (IoU). The principle of IoU is shown in the Figure 3.4.

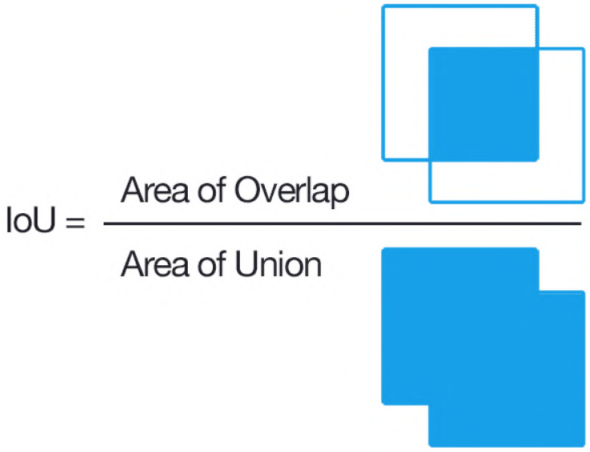


Figure 3.4: Principle of IoU [40].

YOLOv3 is fast deep learning model suitable for applications, where real-time detection and classification is needed. We can observe a huge drop of performance with increased IoU threshold. This indicates that model has problem with perfect fit of predicted bounding boxes and ground truth boxes. Small objects can be detected more precisely than in previous models, due to introduction of mutli-scale predictions.

Implementation of the YOLOv3 model, which we proposed to use in the detection and classification part with its variants was used from [41]. Train, test and inference scripts with computation of processing speed were available. Our datasets that we created for training the detection models were in VOC format. Creation and annotation process of dataset is described in chapter 4 with more details. We needed to transform the coordinates of annotated objects from VOC to YOLO format. Equations 3.2 show, how to convert VOC format of dataset to YOLO format, where b_{cx}, b_{cy} are center coordinates of bounding box and b_w, b_h are width and height of bounding box respectively in YOLO format. Bounding box in VOC format is represeneted by top left coordinates of bounding box $b_{x_{min}}, b_{y_{min}}$ and bottom right coordinates $b_{x_{max}}, b_{y_{max}}$.

$$\begin{aligned} b_{cx} &= \frac{\frac{b_{x_{max}}+b_{x_{min}}}{2}}{image_{width}}, \\ b_{cy} &= \frac{\frac{b_{y_{max}}+b_{y_{min}}}{2}}{image_{height}}, \\ b_w &= \frac{b_{x_{max}} - b_{x_{min}}}{image_{width}}, \\ b_h &= \frac{b_{y_{max}} - b_{y_{min}}}{image_{height}} \end{aligned} \tag{3.2}$$

Second deep learning based method, which we propose to use for tackling the object detection and classification of floating debris problem, is Single Shot MultiBox Detector (SSD) [42]. SSD uses single network for detection and classification similar to YOLOv3. Therefore, it belongs to category of one stage detectors. Proposed deep learning model uses information from feature maps with different resolutions. In comparison with two stage detectors, SSD eliminates regional proposal and feature resampling stage. This elimination of extra steps can remove computational overhead and thus making model faster. Performance of the model in terms of accuracy is comparable to the two stage detectors. Proposed model is anchor/prior based, which means that output consists of predictions of bouding box offsets to priors and class predictions.

SSD is fully convolutional network, it consists of three parts. We chose to use SSD300 with input resolution of 300x300. Reason is that lower resolution will be faster and difference in performance of higher resolution which is 512x512 is only $\sim 2\%$. First part of the model serves as a backbone and it produces low level feature maps. Well known model, which has high performance on image classification task called VGG16 [43] pretrained on ImageNet is used. Use of high performance pretrained model as feature extractor can help with faster training and it can capture the basic information from image. Small adjustments to the model needs to be done before applying it as backbone. Convolutional layers remain, but layers responsible for classification are removed and replaced by the convolutional layers.
Another part of the model architecture are auxiliary layers added on top of the backbone network. Added convolutional layers provide more features maps and they decrease in size, which enables multi scale detection. Architecture of SSD is shown in Figure 3.5.

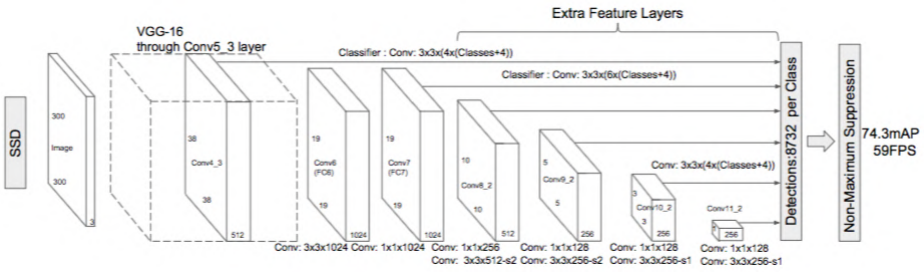


Figure 3.5: Architecture of the SSD model [44].

Object in the image can have many shapes and sizes, therefore we need to discretize the space of potential object occurences. We use priors for the discretization. Priors are used to approximate the shape of bounding box predictions and their size is precalculated. Different possible locations of object in image are tackled by placing priors into every cell in the feature map. Every feature map will have priors with different aspect ratios. Total number of priors defined in the model SSD300 is 8732.

Priors are starting point for predictions of bouding boxes. The goal of the model in terms of predicting locations of bounding boxes is to predict deviation between prior and the bounding box. Equations 3.3 [45] express computation of deviation between prior and bounding box in center-size coordinates. Where c_x, c_y, w, h are coordinates of ground truth bounding box and $\hat{c_x}, \hat{c_y}, \hat{w}, \hat{h}$ are coordinates of the prior. Offsets are normalized by dimension of the prior.

$$\begin{aligned} g_{cx} &= \frac{c_x - \hat{c_x}}{\hat{w}}, \\ g_{cy} &= \frac{c_y - \hat{c_y}}{\hat{h}}, \\ g_w &= log\left(\frac{w}{\hat{w}}\right), \\ g_h &= log\left(\frac{h}{\hat{h}}\right) \end{aligned} \tag{3.3}$$

To obtain predictions of the offsets and scores for classes for each prior, two convolutional layers are needed for each feature map. This is considered as a third part of the model network. Kernel with size 3x3 is used in both cases. Four filters are used to encode offsets. To encode scores for the classes, there should be the same amount of filters as number of classes we want to predict. Kernels are placed on each of the cell of the feature map.

In SSD we determine overlap between predicted bouding boxes and ground truth bounding boxes with IoU 3.4 similar to YOLOv3. Predicted bounding boxes with overlap greater than 0.5 are considered as positive matches, the other as negative matches. During the training, hard negative mining technique is used to tackle the inbalance between positive and negative matches. In hard negative mining negative matches are sorted by highest confidence loss. After sorting, we choose the negative matches with ratio 3:1 at most between negative and positive matches.

In terms of performance, SSD can localize objects more precisely than compared two stage detector. Robustness of the model to different sizes and shapes of objects come from using multiple feature maps. Lack of performance is seen with smaller objects. Problem related to performance of model with different objects sizes can be partially solved by introducing data augmentations. Introduced data augmentations for tackling different sizes of objects are zoom in and zoom out augmentations. Not only detection of different sizes of objects is improved, but also accuracy of whole detection performance is increased.

For the SSD detection model based on deep learning, we used the imple-

mentation available on [45]. We developed script, which is able to use SSD model for detection and classification of floating debris from video source. Within the developed script, measure of FPS is also implemented. For the evaluating the classification ability of our proposed model, implementation of multi-class confusion matrix [46] was needed. As an confusion matrix class, implementation from [47] was implemented to evaluation code of the SSD model.

3.3 Tracking of floating debris

For the grasping of floating debris from water surface we need to detect and also track the floating objects. With the help of tracking, we can later extract the position of tracked objects and plan the path for the UAV to remove them from water. In this section, our solution for solving the challenge of visual multi-object tracking of floating debris on water surface will be presented. Our motivation was to propose method, which can be able to track multiple objects and do not add more computational load to our first part of detection-tracking system, which is based on deep learning. This is due to making our detection-tracking system suitable for the implementation on the UAV hardware. Proposed method does not need to have ability to deal with long term occlusions, since the UAV will fly on top of the floating debris.

Method, which we propose for multi-object visual tracking is Simple online and real-time tracking (SORT) [48]. It is lighter version of DeepSORT [49]. Method is based on nowdays used tracking by detection paradigm. Performance of this type of methods strongly depends on the detections in the input. If the detector can not provide accurate detections, tracking algorithms also fails in performance. So with change of the detector for more accurate one, we can observe improvement of the tracking performance. SORT algorithm claims to be 20x faster than other state of the art tracking algorithms and runs on 260 FPS. During the development, the main focus was on the speed of the algorithm. The reason behind this is that the algorithms used in autonomous vehicles and robotis need to perform fast and in real-time. Proposed method belongs to category of online trackers. Online trackers do not have detections from future frames like offline trackers. They use only informations obtained from previous and current frame for adjusting the tracking. Proposed algorithm consists of three parts. First part is dealing with motion estimation of targets and propagating their position to the next frame. Another part tackles the data association between new detections and tracked targets. Last part is responsible of managing the life span of trackers.

Input to the SORT algorithm are bounding boxes from detector. Size and position of bounding boxes are only used for motion estimation and data association. Estimation model uses the information from bounding boxes to propagate their position to the next frame. Constant linear velocity model is used for approximating the displacement of tracked targets between frames. State of the each target is modeled by horizontal and vertical pixel location, scale, aspect ratio and derivations of the first three mentioned. Tracked target state is updated by associated bounding box and velocity is solved optimally with help of Kalman filter. Linear velocity model is used, when no bounding boxes are associated with tracked targets.

In data association part of the proposed alghorithm, new detections are matched with tracked targets. Tracked targets bounding boxes are predicted to current frame. Assignment cost matrix is determined by computing IoU 3.4 between each bounding box comming from detector and predicted bounding boxes in current frame. IoU_{min} threshold is used to accept assignments that are above the set threshold. For the optimal assigment, Hungarian algorithm is used. Experiments showed that the problem with short term occlusions can be tackled by data association part.

Life span of the tracked objects are handled in third part of the alghorithm. Tracking algorithm assigns the unique identity to each tracked target. Proposed algorithm can not handle cases when the object leaves the frame and re-enter after a few number of frames. This lacking ability is also called re-identification. Therefore if the object leaves the frame, this unique identity is destroyed. Each bounding box from detector with less overlap than IoU_{min} to targets bounding boxes is considered as new target. New target tracker is initialized with the size and position of its bounding box and zero velocity. The velocity component covariance of this new tracker is assigned with high values. Threshold min_{hits} which express how many assignments to the new tracked target needs to be accepted from detections before considering tracker as true positive is introduced. Another T_{Lost} threshold is presented. If the trackers are not matched with detections for T_{Lost} frames, then tracker is terminated. If the high T_{Lost} threshold is set, unbounded localization error due to incorected trackers can be observed.
Every presented threshold in proposed method needs to be experimentally tuned to specific application. SORT algorithm was evaluated on MOT benchmark dataset where videos with moving and static camera are used. Experiments show that proposed method can be compared in performance also with offline methods. A balanced trade off between speed and accuracy can be observed.

We created scripts that implement SORT algorithm for tracking the floating debris on water surface with each of the proposed detection models. These

detection models provide detections in form of bounding boxes and classes of detected floating debris to the input of the SORT algorithm. Authors of the SORT made its implementation available for further research [50]. We modified the code that every tracker is able to store the information about class of the tracked target. We used tool available on [51] for the evaluation of the tracking pipeline. We can see the pseudocode 1 of the SORT algorithm below.

Algorithm 1: Pseudo code for tracking of floating debris in video.

Input: Video
Output: Tracked floating debris on water surface in video
Initialization of deep learning model used for detection;
Initialization of SORT tracking algorithm (IoU_{min}, T_{Lost}, min_{hits});
while *Frames available* **do**
    Read current frame;
    Obtain detections with classes from detection model;
    if *number of detected objects > 0* **then**
        Update SORT algorithm with new detections;
    else
        Update SORT algorithm with empty detections;
    end if
end while

Chapter 4

Dataset

Over the last few years, the saying that the data has a value of gold, expanded in the machine learning community. Every machine learning and deep learning approach needs to have a large amount of data on the input. Input data are used for training and evaluating the performance of the models. The more different data we have, the more accurate and robust can our models be. In this chapter we present our custom dataset of floating debris on water surface, which we collected from river and pond. We had two main reasons for collecting our dataset. The first reason was a lack of annotated datasets of floating debris taken from UAV available online. Secondly we wanted to bring the added value to the research community.

4.1 Data split and acquisition

The foundation of every computer vision deep learning based approach are images and videos. Our dataset consists of images collected in different weather conditions, day times and water conditions. First collection of the dataset was performed in sunny weather without wind on the pond. Sun was directly over the water and that brings challenges to the dataset, which will be discussed later. Another dataset was gathered in partially cloudy and windy weather on the river.
Floating debris objects consisted of three classes, which were plastic menu boxes, plastic bottles and plastic bags. To prevent sinking of the debris, every piece was tied up to tiny rope in a random order. Every rope from dataset collected on the pond was tied also with the pier. Ropes from the river dataset

were not tied to anything, therefore images are without a specific distractor, like pier which is visible in images in previous mentioned dataset. We were trying to arrrange the floating debris as groups and also as individual objects, because in real maritime environment floating debris also occurs in this form.

Datasets were gathered by using UAV equiped with two cameras facing down. UAV that was used for collection of datasets is shown on Figures 4.1, 4.2 and 4.3. First camera used for collection of a dataset was Intel RealSense d435i with frame rate of 30 FPS. Another camera, which we used, was Basler dart with frame rate of 60 FPS. Each of the photo captures zero or more floating debris objects. Images from first dataset collection was captured by Intel RealSense camera with resolution of 1280x720. Dataset from the river consists of images taken by Basler dart camera with resolution of 1600x1200.

Figure 4.1: Image capturing Intel RealSense camera mounted on the UAV.

Figure 4.2: UAV used for collection of datasets.

Figure 4.3: Image capturing Basler dart camera mounted on the UAV.

Images were split into Train, Test 1 and Test 2 datasets. Train and Test 1 datasets consist of images from dataset, which was collected on the pond. The difference between Train and Test 1 datasets was in adding new floating debris and the order of tied objects to the rope differs. Test 2 dataset consists of images collected only from Basler dart camera. These images were captured on the river and they are considered as more difficult than images from Test 1 dataset. Examples of the images from the datasets can be seen on the Figure 4.4.

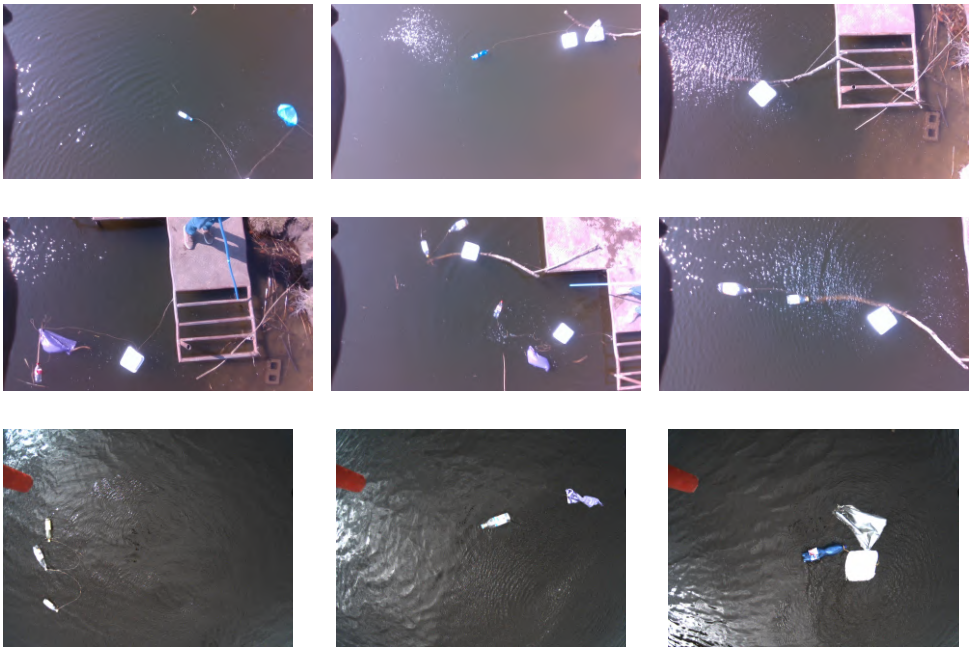


Figure 4.4: Train images - top row, Test 1 images - middle row, Test 2 images - bottom row.

Every image from train and test datasets had to be annotated. For the image annotation process, we used a tool called LabelImg [52] to draw bounding boxes around floating debris and assign one from three classes to them. Annotations are in VOC format and they are represeneted by top left coordinates of bounding box and bottom right coordinates.
From each of the test datasets, we created the video by using FFmpeg [53] tool. Video was annotated by CVAT [54] annotation tool, which produced annotations in MOT format. Each floating debris object in video was annotated by one class called floating debris. We applied a rule, that every object with visibility less than 50% in the image, will not be annotated. This rule was applied to annotation of images and videos.

4.2 Statistics of dataset

Recorded files which were selected to serve as training and testing datasets consisted of totally 36 000 frames. UAV flight altitude during the data collection was 1.7m - 5.4m from the take off point. From the files for Train and Test 1 dataset, we extracted every 10th frame. For the Test 2 dataset we extracted every 15th frame from recorded files. After extraction of frames, we manually removed the frames, which did not contain floating debris. The rest of extracted frames were annnotated. Videos in our custom dataset were created from files that were used to create test datasets. From the raw frames collected from the river, we created Test 2 video with use and annotation of every 3rd frame. All frames from Test 1 dataset collected on the pond were used for video creation.
Total number of extracted frames in our dataset can be seen in the Table 4.1.

	Train	Test1	Test2
number of frames	1104	319	320
	Test1 video	Test2 video	
number of frames	4572	7812	

Table 4.1: Number of annotated frames used in the datasets and Test videos.

In the table 4.2 we can see the number of objects per class in every dataset.

	Menu boxes	Plastic bags	Plastic Bottles	Total objects in dataset
train	963	591	944	2498
test1	254	186	343	783
test2	142	300	412	854

Table 4.2: Statistics of objects per class in each dataset.

Distribution of labeled objects per frame is shown in Figure 4.5.

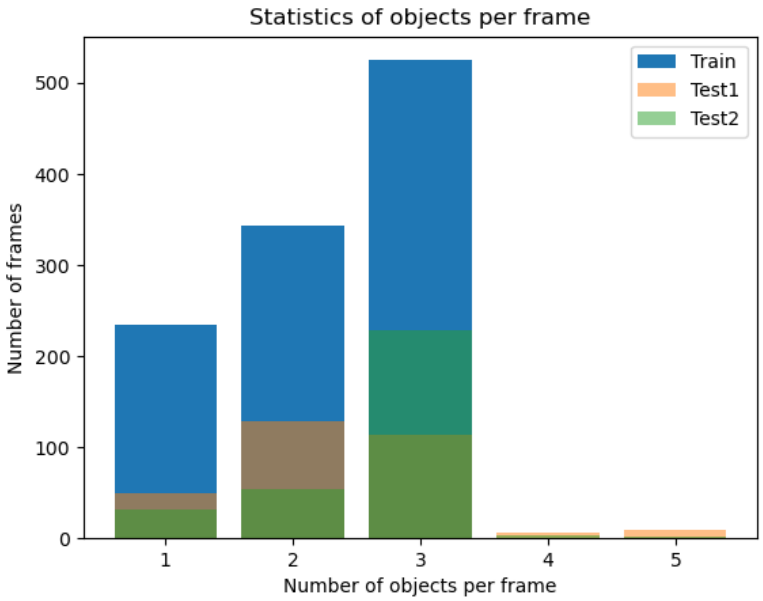


Figure 4.5: Statistics of labeled objects per frame.

4.3 Challenges in dataset

Since the nature is unpredictable, many challenges of gathering datasets in real world conditions can arise. Size and shape of floating objects are changing over time. First challenge that we observed during collection of our custom datasets were overexposed images. Overexposure was caused by strong reflection of light from the water surface. That may cause the missed detection of floating debris. Another problem emerged with the changes in shape of plastic bags. When the wind was present, plastic bag expanded in its size. On the other hand , when the bag was calmly floating on water surface, due to water flow the size shrunk. During the presence of wind, small waves appeared and caused the light floating debris to submerge and resurface later. Water started to fill the plastic menu boxes when they were present in a water for a long time. Filled menu boxes with water were submerged by half of their size. When the UAV was flying under 3.5 meters, changes in motion and shapes of floating debris were observed due to wind caused by the rotors. From the high UAV flying altitude objects, which were in clusters, visually appeared like one floating object. Another case when the floating debris appeared like one object, was when the strong reflection from the menu

box was present near other floating debris. Changes in the size and shape of the floating objects were observed, when the UAV tilted during the flight. Examples of the images containing the challenges in the dataset can be seen on the Figure 4.6.

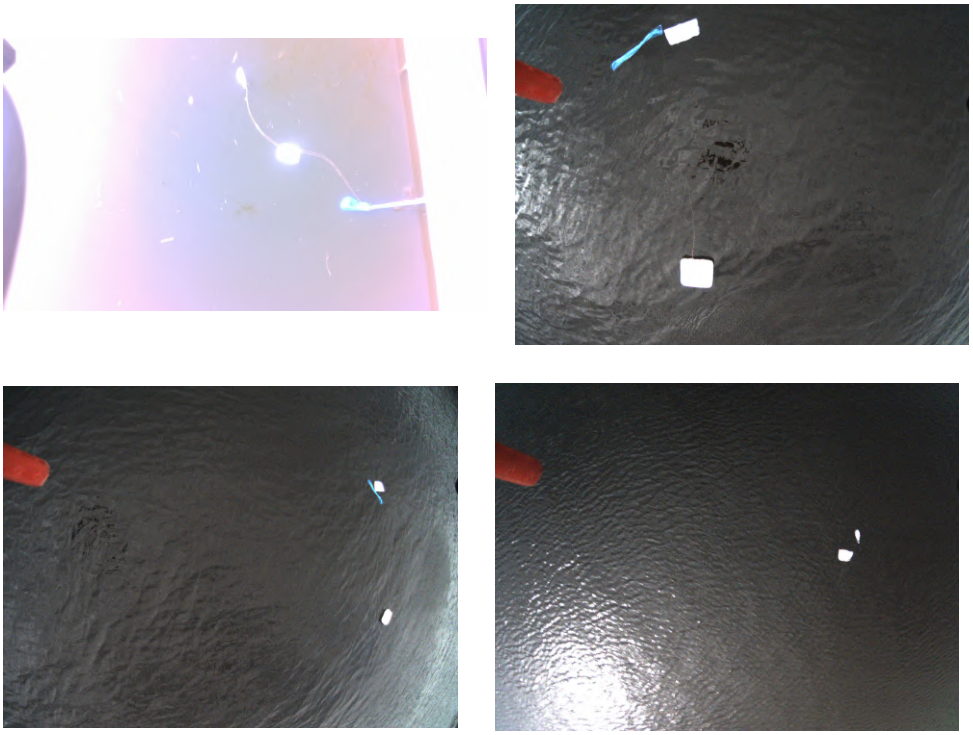


Figure 4.6: Top left image shows overexposure. Top right image shows change in the shape of plastic bag. Bottom left image shows captured image when drone is tilted. Bottom right image shows more objects appearing as one.

Chapter 5

Experiments and Results

In this chapter we will present experiments and results. Experiments were performed on proposed detection models based on deep learning and proposed tracking algorithm. First we show experiments on each of the detection model separately. Then we experiment with the proposed tracking algorithm implemented with each of the detection model. Results from the performed experiments will be analyzed. After the analysis of the results, we will propose combination of detection model with tracking algorithm, which will be the most accurate and fast enough, based on our experiments.

Experiments on the detection models based on deep learning consist of two parts. First part is dedicated to experimentation with hyperparameters of the model during the training stage, for example learning rate, optimizer, number of epochs and so on.
Another part was dedicated to experiment with confidence score of the model and IoU thresholds. Confidence score can tell us, how is model confident with the provided detection. In other words, how is model sure that the object is on the predicted place. In order to not waste the energy consumption of the UAV, we need to send it only on places, where we are sure that the floating debris is present. If we send the UAV on the place with no floating debris, the energy will be wasted. IoU can tell us, how accurate can model detect the floating debris. In other words, how big is overlap between ground truth object on the image and predicted bounding box. We wanted our models to be at least 50% confident and have an overlap between ground truth objects at least 50%. We explored combinations of confidence and IoU thresholds of 50% and 75%.
Another experiments dedicated to initialization of proposed tracking algorithm were performed. Speed will be measured in each of the before mentioned

experiments in FPS. All the experimnets were performed on hardware with NVIDIA GeForce GTX 1050 graphic card.

Our detection models were not trained from scratch. For the training, technique called transfer learning [55] was used. Used technique allows to apply learned informations from previous training in similar domain to the new specific domain. In our case it means to first iniliaze feature extractor with pretrained weights on ImageNet like in SSD model case or iniliaze whole model like YOLOv3. After initilization of weights, we need to train the model by number of epochs to learn domain specific informations. This type of training is also called fine tuning. With the use of transfer learning, training time to new domain can be shortened and also smaller dataset is needed. Proposed models will be evaluated by mean average precision (mAP) metric [56] and multi class confusion matrix will be constructed, to get more information about classification task of our models [46]. Evaluation of tracking algorithm implemented on top of the detection models will be evaluated by the classic MOT metric [57].

SORT tracking algorithm had parameters that needed to be initilized. After experimentation with the parameters we set the IoU_{min} threshold to 50%. Minimum hits parameter, which is responsible of counting the association between detections and new created trackers, is set to 2. Last parameter that needed to be initialized was T_{Lost} parameter and it was set to 1. This parameter is responsible of checking for how long is tracker without correction. Value 1 was set to prevent the unbounded localization error. Tracking algorithm was tested on Test 1 and Test 2 videos.

5.1 SSD model

First proposed detection model was SSD. Final model was trained for 10 000 iterations with batch size of 16. For the optimizer, stochastic gradient descent was used with initial learning rate of 0.001, momentum 0.9 and weight decay of 0.0005. Learning rate was decayed after 7 000 and 9 000 iterations by multiplier of 0.1.
During the training stage, we used data augmentations applied to training data. These augmentations help to extend the dataset with new and different examples of images. Another benefit of data augmentations is an improvement of the model performance and robustness. Augmentations that we used were random horizontal flip, zoom in and zoom out operations and photometric distorstions. Distortions consisted of changing the image contrast, saturation, hue and brightness in random order.

During the testing stage, the model was initialized to keep top 200 detections. Value for IoU threshold in Non Maximum Suppression was set to 0.45.

In the table 5.1, we can see that the model achieves best mAP of 84% and 33.9% in Test 1 and Test 2 dataset respectivelly by confidence score and IoU thresholds set to 50%. The second highest achieved mAP is with increased threshold of confidence score to 75% with remaining IoU threshold. For the Test 2 dataset we can see significant drop in performance of the model. Model was not robust enough to adapt on new data. Test 2 dataset had many challenges, like varying shapes of floating debris due to waves, different color of water and different exposure. We can observe that class menu box had the highest AP among all presented classes in Test 1 dataset with all combinations of thresholds. In the Test 2 dataset menu box achieved also the highest AP among all classes, except the thresholds of confidence and IoU set to 50%, where plastic bottle class achieved the highest AP. Difference between datasets is also in the AP of class plastic bag which had the lowest AP in Test 2 dataset in comparison with Test 1 dataset. This was caused by mentioned challenges in the dataset.

Test 1 dataset				
Conf / IoU	0.50 / 0.50	0.50 / 0.75	0.75 / 0.50	0.75 / 0.75
Menu box AP	0.905	0.811	0.815	0.811
Plastic bag AP	0.894	0.791	0.807	0.712
Plastic bottle AP	0.720	0.585	0.631	0.509
mAP	0.840	0.729	0.751	0.677
Test 2 dataset				
Conf / IoU	0.50 / 0.50	0.50 / 0.75	0.75 / 0.50	0.75 / 0.50
Menu box AP	0.381	0.307	0.324	0.307
Plastic bag AP	0.232	0.089	0.163	0.089
Plastic bottle AP	0.403	0.198	0.259	0.151
mAP	0.339	0.198	0.249	0.182

Table 5.1: Evaluation of SSD model on test datasets.

By construction of confusion matrices we can have more information about classification ability of our models. Confusion matrices on the Figures 5.1 and 5.2 were constructed with confidence and IoU thresholds set to 50%, which corresponds to the highest obtained mAP on the both test datasets. Plastic bottle class was classified without any misclassification between target classes, but had the highest count of missed detections in both test datasets. Menu box class had the lowest count of false positive detections in both datasets.

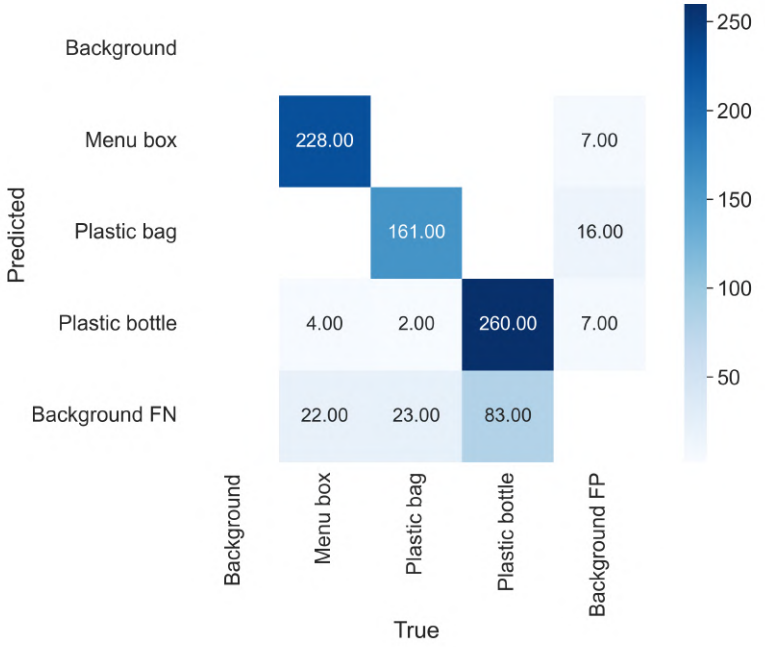


Figure 5.1: Confusion matrix SSD Test 1 dataset.

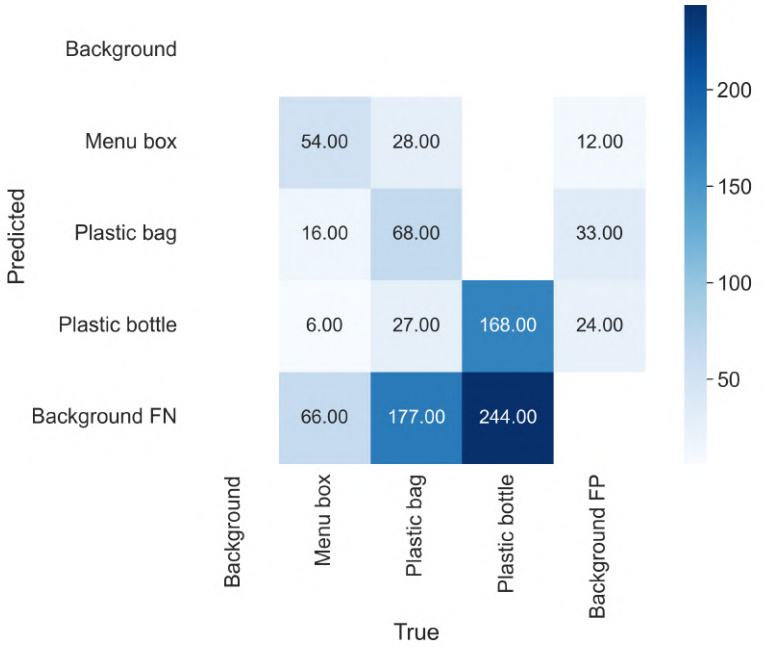


Figure 5.2: Confusion matrix SSD Test 2 dataset.

Experiments with SORT tracking algorithm on top of the SSD detection model were gathered. During evaluation of tracking algorithm was detection model initilized to keep top 200 detections. IoU thresholds for Non Maximum Suppression was set to 45% and confidence threshold was set to 50%.
In the Table 5.2 we can see results from testing the tracking algorithm using

detections from SSD detection model. We can see significant drop of the MOTA in the Test 2 video. Since the SORT tracking algorithm is highly dependent on detection model performance, observed drop in MOTA is due to low performance of SSD model. IDSw increased almost three times in Test 2 video. MT decreased and high increase was observed with ML trajectories. MOTP remained almost the same with deacrease of only 4%.

	MOTA	MOTP	MT	PT	ML	Frag	IDSw
Test 1 video	69.76	78.16	34	25	3	259	176
	MOTA	MOTP	MT	PT	ML	Frag	IDSw
Test 2 video	26.46	74.73	4	44	58	412	476

Table 5.2: Evaluations of SORT using detections from the SSD model.

5.2 YOLOv3 model

Another detection model based on deep learning, which we proposed was YOLOv3. Final model which we decided to use was trained for 30 epochs. Image size that we used for training was 416x416. Stochastic gradient descent was used as optimizer with initial learning rate of 0.01. Momentum of the optimizer was set to 0.937 and weight decay to 0.0005. During the training 3 warm up epochs were used with warm up momentum of 0.8 and learning rate 0.1. Data augmentations, which we used to expand the training data examples, consisted of HSV hue and saturation, image rotation and translation was used with image scale and shear. More about augmentation and hyperparameters details can be found in the files, which are part of the appendix.

In the table 5.3 we can see that the model achieves best mAP of 96.4% and 70.2% in Test 1 and Test 2 dataset respectivelly by confidence score and IoU thresholds set to 50%. The second highest achieved mAP is with increasing threshold of IoU to 75% with remaining confidence threshold. For the Test 2 dataset, we can see that the model can adapt to new data better than SSD model. We can observe that class menu box had the highest AP among all presented classes in Test 1 dataset with all combinations of thresholds. Plastic bottle has the highest AP among all classes in the Test 2 dataset, except the combinations of confidence and IoU threshold both set to 75% where plastic bag achieves the highest AP. We can see that between the lowest and highest thresholds combinations there is only 3.5% drop in performance in case of Test 1 dataset and 5% drop in case of Test 2 dataset. Plastic bag has the lowest AP with almost all combinations of thresholds. In comparison with SSD model, increase in mAP by 12.4% and 36.3% in Test 1 and Test 2 datasets are observed.

Test 1 dataset				
Conf / IoU	0.50 / 0.50	0.50 / 0.75	0.75 / 0.50	0.75 / 0.75
Menu box AP	0.990	0.988	0.989	0.986
Plastic bag AP	0.922	0.914	0.860	0.852
Plastic bottle AP	0.979	0.960	0.963	0.949
mAP	0.964	0.954	0.937	0.929
Test 2 dataset				
Conf / IoU	0.50 / 0.50	0.50 / 0.75	0.75 / 0.50	0.75 / 0.75
Menu box AP	0.646	0.646	0.611	0.611
Plastic bag AP	0.590	0.550	0.541	0.803
Plastic bottle AP	0.870	0.822	0.841	0.541
mAP	0.702	0.673	0.664	0.652

Table 5.3: Evaluation od YOLOv3 model on test datasets.

After evaluation of mAP of the YOLOv3 detection model, we created confusion matrices to obtain more detailed information about classification ability of the model. Confusion matrices on the Figures 5.3 and 5.4 were constructed with confidence and IoU thresholds set to 50%, which coresponds to the highest obtained mAP on the both test datasets. As we can see that menu box class had no missed detections in the Test 1 dataset. Plastic bag class had the lowest count of false positive detections among all classes in the both test datasets. No misclassification between plastic bottle and target classes in Test 2 dataset was observed, same as in the SSD model case but with lower count of missed detections.

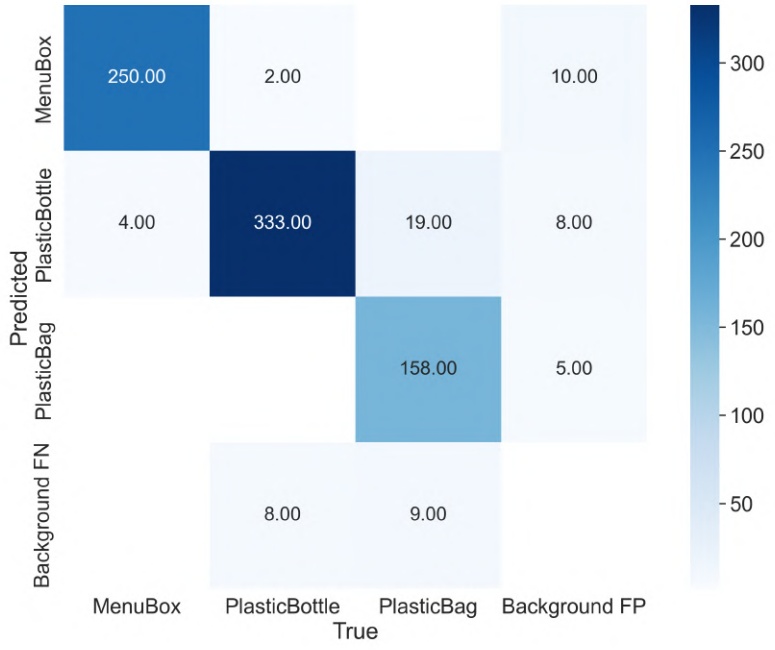


Figure 5.3: Confusion matrix for YOLOv3 Test 1 dataset.

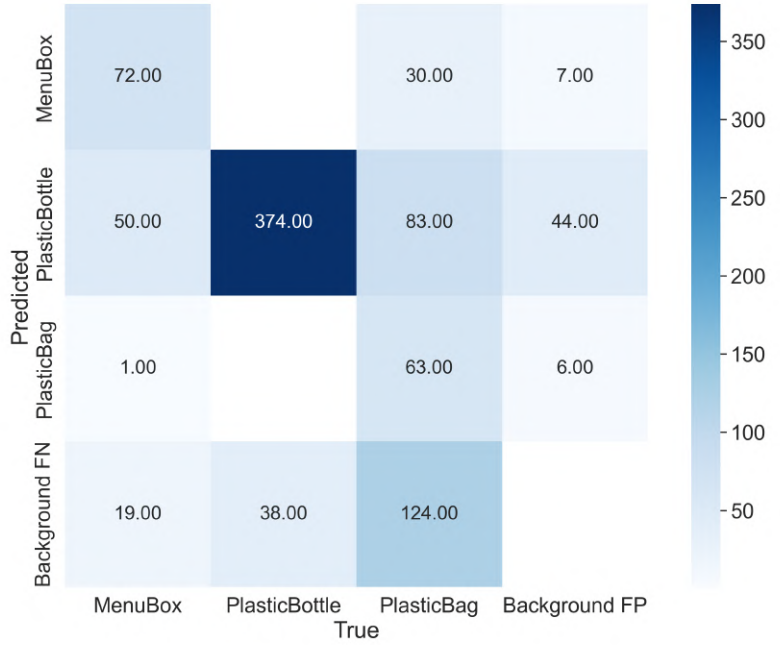


Figure 5.4: Confusion matrix for YOLOv3 Test 2 dataset.

Experiments with SORT algorithm on top of the YOLOv3 detection model were performed. YOLOv3 parameters for inference were set same as in the SSD model case. In the table 5.4 we can observe the drop of the MOTA by 33% between Test 1 and Test 2 video. The drop is not as high as it was in the case of SSD model. When SSD model was used to provide detections, drop in MOTA was almost 45%. In case of Test 1 video ID switches lowered by half in comparison with using SSD model as a base detector. ML are lowered also by half and significant increase of number MT tracks in Test 2 video is shown, if the detections from YOLOv3 are used.

	MOTA	MOTP	MT	PT	ML	Frag	IDSw
Test 1 video	85.08	81.06	48	12	2	152	84
	MOTA	MOTP	MT	PT	ML	Frag	IDSw
Test 2 video	52.70	79.93	33	51	22	511	414

Table 5.4: Evaluations of SORT using detections from the YOLOv3 model.

5.3 YOLOv3-Tiny model

Another model that we used for the experiments was YOLOv3-Tiny. YOLOv3-Tiny is lighter version of the YOLOv3 model. Model has less convolutional layers and it detects objects only on 2 scales, thus can be faster but less

accurate. Experimenting with hyperparameters during the training, YOLOv3-Tiny has the same configuration of hyperparameters as YOLOv3. Same type of augmentations with different values on the traing set is used and more details can be found in the files which are part of the appendix.

In the table 5.5 we can observe that the model achieves best mAP of 94.5% and 61.8% in Test1 and Test2 respectivelly by confidence score and IoU thresholds set to 50%. Difference between YOLOv3 model is in the second highest mAP, which is achieved for increasing the confidence threshold to 75% with remaining IoU threshold. We can observe that the adaptation of the model to Test 2 dataset is better than SSD model by almost 30%. In Test 1 dataset plastic bag has the second highest AP in comparison with YOLOv3, where plastic bag had the lowest AP among all classes. Plastic bottle has highest AP among all classes in Test 2 dataset like in YOLOv3, but the difference in exception where another class achieves higher AP is in menu box class instead of plastic bag class, where the confidence and IoU thresholds are set to 50% and 75% instead of 75% and 75%. In case of YOLOv3-Tiny, plastic bag achieved the lowest AP in Test 2 dataset.

Test 1 dataset				
Conf / IoU	0.50 / 0.50	0.50 / 0.75	0.75 / 0.50	0.75 / 0.75
Menu box AP	0.977	0.959	0.972	0.955
Plastic bag AP	0.975	0.800	0.881	0.744
Plastic bottle AP	0.882	0.729	0.766	0.669
mAP	0.945	0.829	0.873	0.789
Test 2 dataset				
Conf / IoU	0.50 / 0.50	0.50 / 0.75	0.75 / 0.50	0.75 / 0.75
Menu box AP	0.591	0.560	0.503	0.503
Plastic bag AP	0.504	0.337	0.477	0.385
Plastic bottle AP	0.760	0.526	0.706	0.539
mAP	0.618	0.474	0.562	0.476

Table 5.5: Evaluation od YOLOv3-Tiny model on test datasets.

Confusion matrices on the Figures 5.5 and 5.6 were constructed also in the case of YOLOv3-Tiny evaluation and they were constructed with the combination of thresholds corresponding to the highest mAP. From the figures we can observe that there is no class with any misclassifications comapred to the previous cases with SSD and YOLOv3 models. In comparison with YOLOv3 model, menu box has the lowest count of false positive detections in both test datasets.

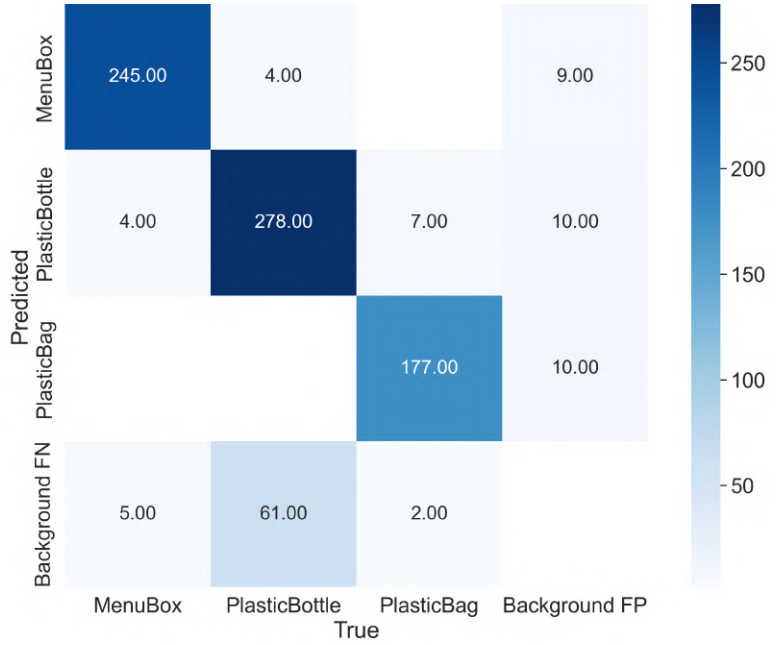


Figure 5.5: Confusion matrix YOLOv3-Tiny Test 1 dataset.

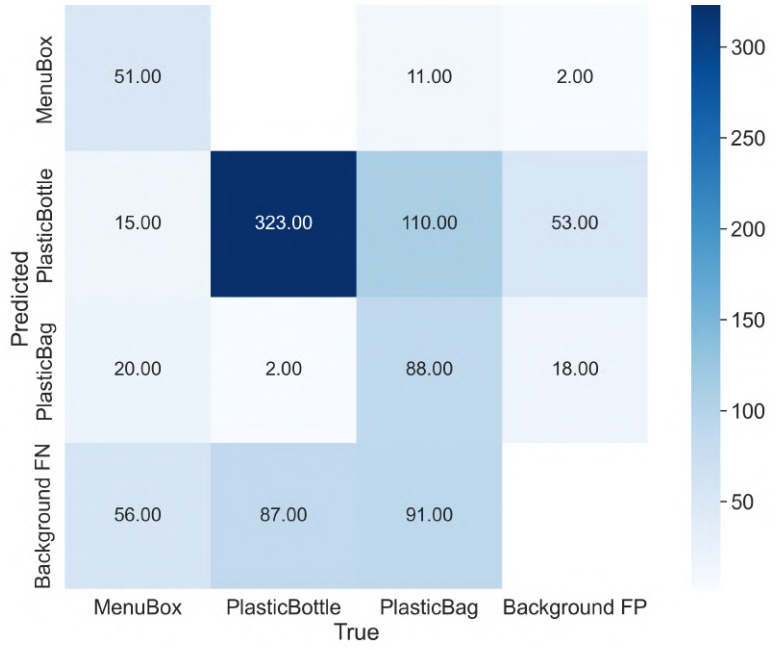


Figure 5.6: Confusion matrix YOLOv3-Tiny Test 2 dataset.

Experiments with SORT algorithm on top of the YOLOv3-Tiny was also performed. All parameters were set as in the case of YOLOv3 model. In the table 5.6 drop in the MOTA by 30% between Test 1 and Test 2 video can be observed. Number of fragmentations of trackers by missed detection is significantly higher than in case, where the detections are provided from

YOLOv3 model. Number of ML is lower by half and significant increase in count of MT tracks is observed in comparison with SORT algorithm on top of SSD detection model in Test 2 video.

	MOTA	MOTP	MT	PT	ML	Frag	IDSw
Test 1 video	75.09	77.632	39	16	7	208	138
	MOTA	MOTP	MT	PT	ML	Frag	IDSw
Test 2 video	44.153	75.203	19	62	25	720	586

Table 5.6: Evaluation of SORT algorithm with detections provided from the YOLOv3-Tiny model.

5.4 Experiments with Lidar

Although only the camera is mentioned in the assignment of this thesis, we performed small experiments with lidar. The main idea of these small experiments, was to find out, if the mounted lidar on the UAV can sense the floating debris. In [58], the usage of lidar for sensing the ice-floes was explored. The goal was to develop hazard warning and avoidance system for ships. Their experimental setting consisted of plastic polygons, which simulated the floating ice-floes in experimental water tank. They observed that laser reflections came only from plastic polygons.

We performed small experiments with the Ouster OS0-128 Lidar. Laser wavelength was 865nm. Lidar, which was used in [58] had laser wavelength of 905nm. Both of these lasers operate in infrared spectrum. In the Figure 5.7, we can see how the Ouster was mounted on the drone. This drone was also used for a collection of datasets. Flight altitude of UAV was 1.5. - 7 m.

Figure 5.7: Image capturing lidar mounted on the UAV.

Lidar can bring complementary information that can improve the detection of floating debris on water surface. Lidar can help to detect floating objects, even in cases where detectors based on deep learning can fail. Since the lidar depth readings are independent on the lighting conditions of water surface, they can detect the floating debris, where the reflection of light is high. Results from our performed experiments are in the next figures.

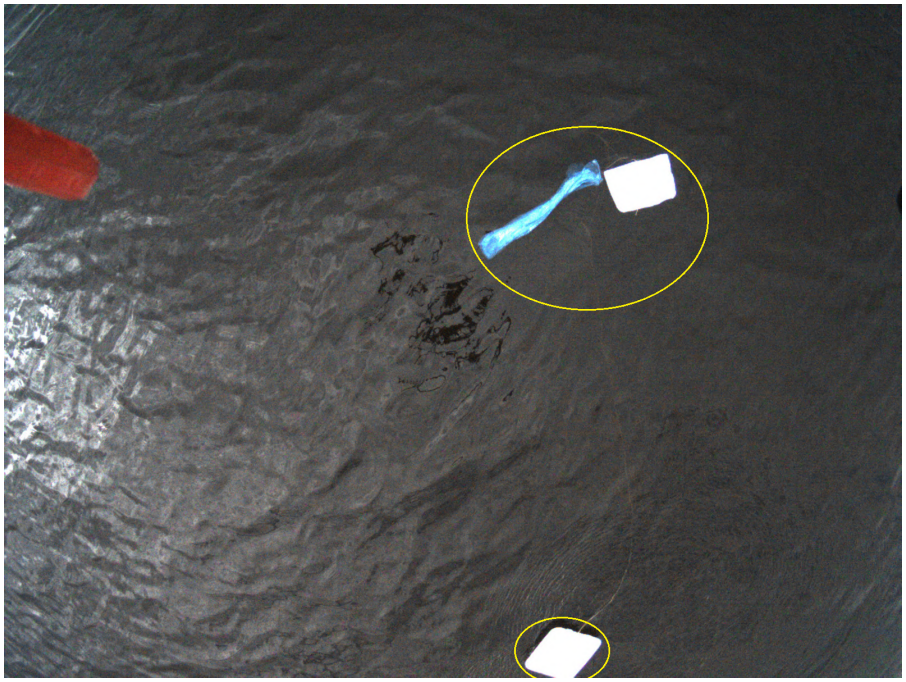


Figure 5.8: Image capturing floating debris.

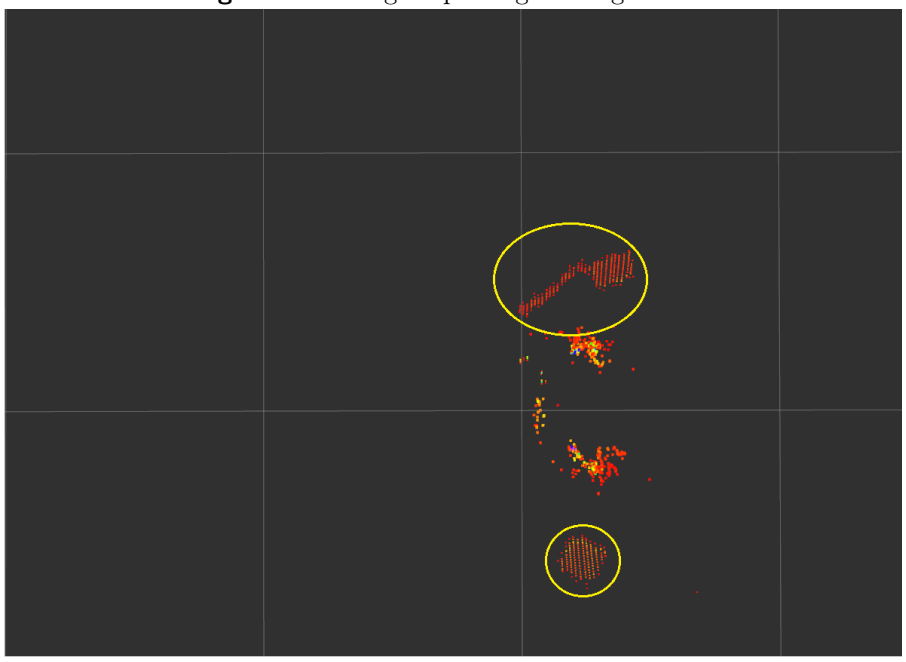


Figure 5.9: Lidar data corresponding to Figure 5.8.

Figure 5.10: Image capturing floating debris.

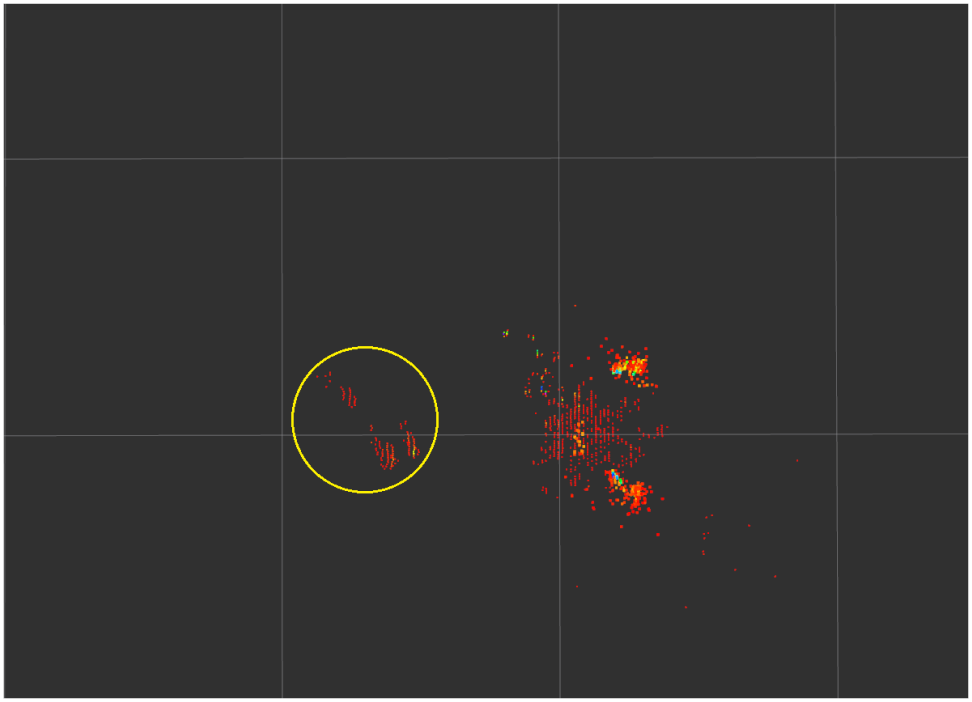


Figure 5.11: Lidar data corresponding to Figure 5.10.

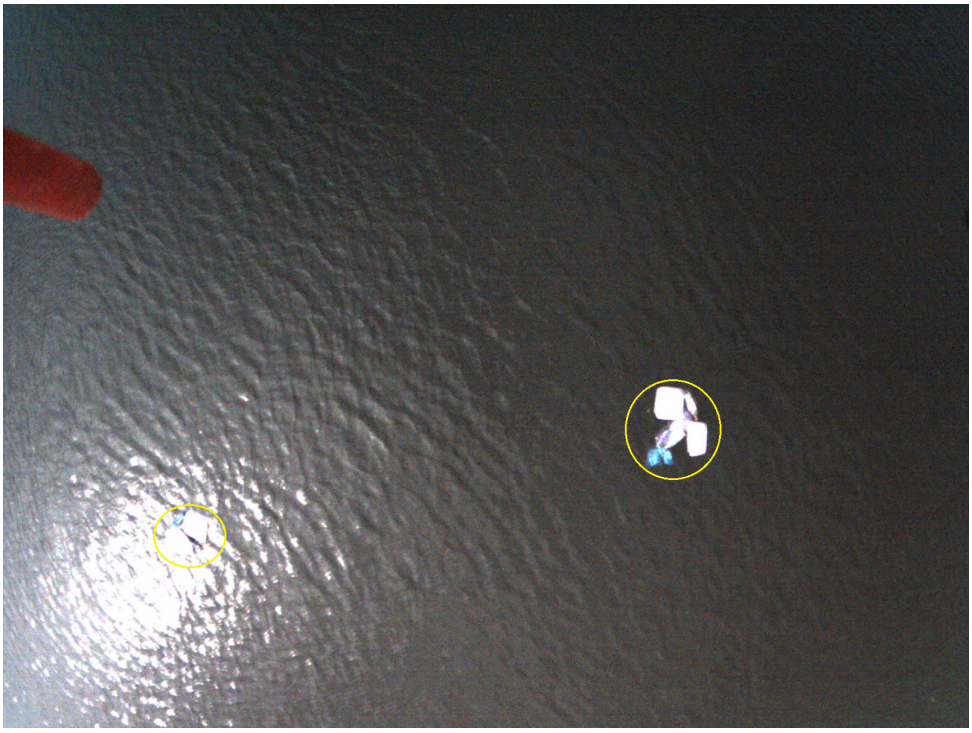


Figure 5.12: Image capturing floating debris.

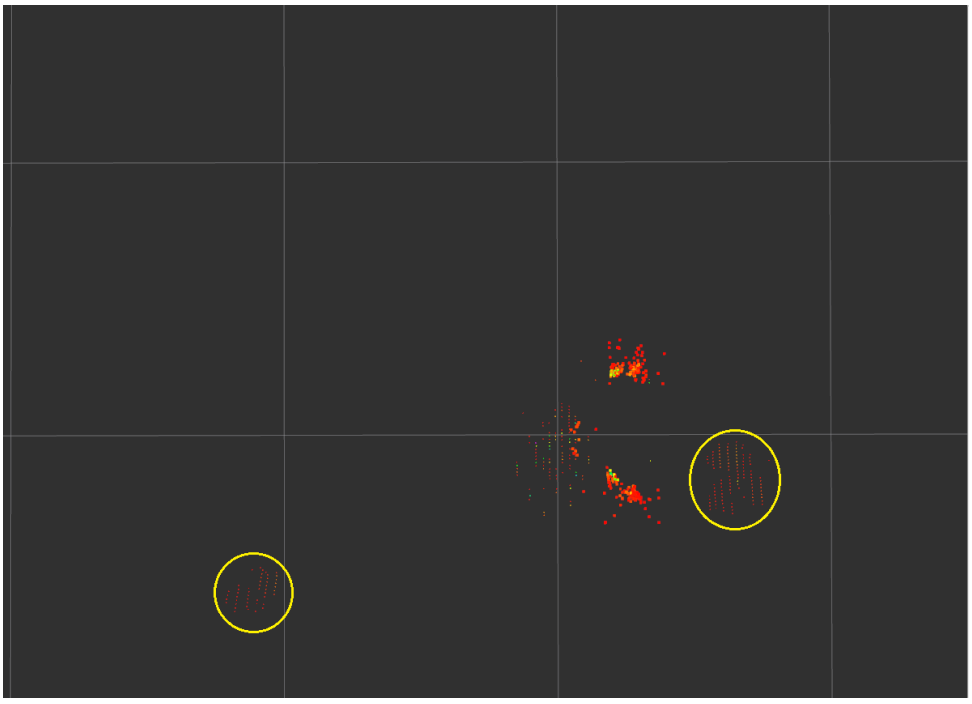


Figure 5.13: Lidar data corresponding to Figure 5.12.

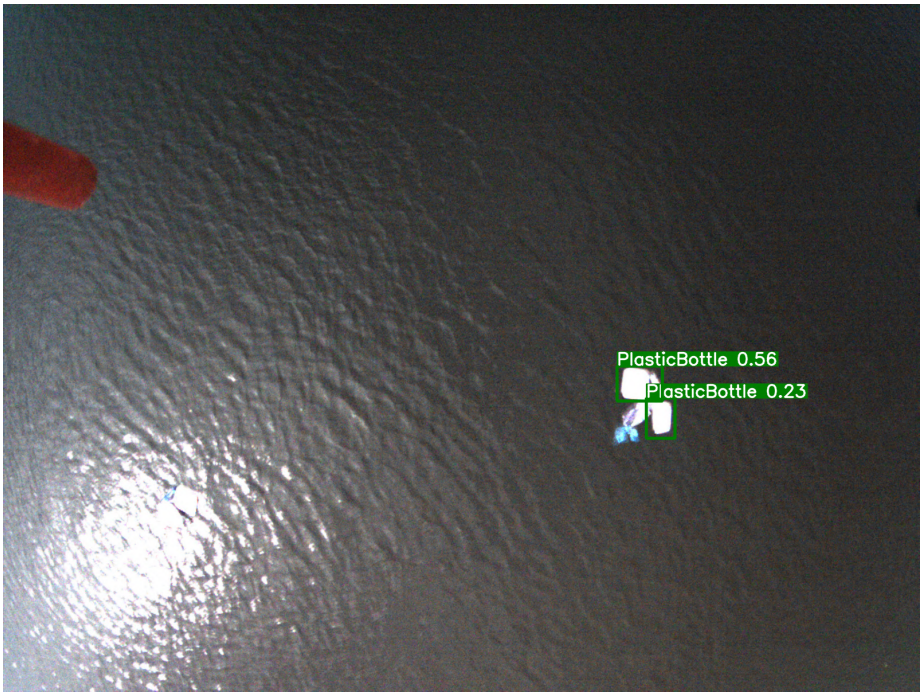


Figure 5.14: Object detection that failed to detect floating debris, where the reflection of light from the water surface was high. Flight altitude was $\sim$ 7 m.

Performed experiments showed that lidar mounted on the UAV can sense the floating debris on water surface. Lidar was able to sense small waves created by rotors of the UAV, when the flight altitude was low. These small waves created a noise in the data. The output from the YOLOv3 model is shown in the Figure 5.14. The model failed to detect floating debris in places, where was high reflection of light from water surface. Flight altitude of the UAV capturing this image was $\sim$ 7m. In the Figure 5.13, we can see the lidar data capturing the same scene where the detection of model failed. The floating debris is sensed by lidar in the place of high reflection of light. This shows that the lidar can provide complementary information in detecting the floating debris, where the object detection models based on deep learning fail.

5.5 Summary

We will summarize our results from evaluation of detection models based on deep learning and tracking algorithm method in this section. Since our detection-tracking system will be used on board of the UAV, speed of the system is important. We evaluated the speed of the each method in terms of how fast can they process frames per second (FPS).
In the Table 5.7 we can see the evaluation of speed of each model separately

and with implemented tracker on top of them. First thing that we can observe is that there is difference in FPS comparing performance of the methods on Test 1 and Test 2 dataset. This is due to Test 2 dataset having higher resolution.
Proposed SSD model has the lowest FPS in comparison with YOLOv3 and YOLOv3-Tiny models. The fastest performing model is YOLOv3-Tiny model with processing 76 FPS for Test 1 dataset and 63 FPS for Test 2 dataset. YOLOv3-Tiny has also the biggest difference in FPS between the two test datasets. If we consider real-time performance of 30 FPS, suitable models will be: YOLOv3-Tiny, which achieves real-time performance and YOLOv3, which achieves almost real-time performance.
Our proposed tracking algorithm aimed to not add computational load to the hardware of the system. In the table we can see that proposed SORT tracking algorithm implemented with the detection models runs on same FPS like detection models without implemented tracking.

	T1 - D	T2 - D	T1 - DT	T2 - DT
SSD	10 FPS	6 FPS	10 FPS	6 FPS
YOLOv3	28 FPS	22 FPS	28 FPS	22 FPS
YOLOv3-Tiny	76 FPS	63 FPS	76 FPS	63 FPS

Table 5.7: Speed evaluation of proposed methods. D - Detection only DT - Detection with tracker.

In the Table 5.8 we have a summary of the highest mAP from both test datasets that the models achieved in our evaluation.

Test 1 dataset				
	Menu box AP	Plastic bag AP	Plastic bottle AP	mAP
SSD	0.905	0.894	0.720	0.840
YOLOv3	0.990	0.922	0.979	0.964
YOLOv3-Tiny	0.977	0.975	0.882	0.945
Test 2 dataset				
	Menu box AP	Plastic bag AP	Plastic bottle AP	mAP
SSD	0.381	0.232	0.403	0.339
YOLOv3	0.646	0.590	0.870	0.702
YOLOv3-Tiny	0.591	0.504	0.760	0.618

Table 5.8: Summary of top achieved mAP for each of the proposed model on both test datasets.

Every proposed detection model achieved the highest mAP with a combination of confidence and IoU thresholds set to 50%. SSD model achieved the overall lowest mAP in comparison with the YOLO models. Especially, we can see the significant drop in mAP for the SSD model in Test 2 dataset which is 50.1%. This shows that model was not robust enough to adapt to the new data. The lowest drop between test datasets was achieved by the

YOLOv3 model. In the table we can see that the menu box class had the highest AP in Test 1 dataset. The plastic bottle had the highest AP in Test 2 dataset, on the other hand plastic bag obtained the lowest AP among all classes in the Test 2 dataset.

In the next figures, we show examples of the outputs from detection models, which were used in the experiments. In the Figure 5.15 we can see the output from the SSD and YOLOv3 model respectively on the overexposed image. YOLOv3 model was able to detect the object, which was missed by the SSD but with an incorrect class. In the Figure 5.16 we can see successful detection with correct classes by the YOLOv3 model.
The correct detections by SSD and YOLOv3 model respectively can be observed in the Figure 5.17. All the objects were detected with correct classes, even with the distractor in the form of the pier. In the Figure 5.18 there are three correctly detected objects, which are plastic bottle, plastic bag and menu box. The output is from the YOLOv3 model.

Comparison between detections from the YOLOv3-Tiny and YOLOv3 model are shown in the Figure 5.19. YOLOv3-Tiny falsely detected a plastic bottle in the place where the wooden branch was present. In the Figure 5.20 we can see that YOLOv3 was not able to detect correctly the floating debris from the height $\sim 7m$. The model also missed the detection of a cluster in the place, where the high reflection of light from the water surface was present. Incorrect detection of a visible cluster is also observed, when the menu box with plastic bottle was detected and classified as plastic bottle. Challenges that occured in the datasets are more discussed in the chapter 4.

SSD model achieved mAP of 84% in the Test 1 dataset and the output of correct detections and classificiations can be seen in the figure 5.21. In the Test 2 dataset, the SSD model performed the worst in mAP among all experimented models. Comparison between detections of SSD and YOLOv3-Tiny model in Test 2 image example is shown in the figure 5.22. SSD model was not able to detect any of the plastic bottles presented in the image, where the lighter version of YOLOv3 performed well with detection and correct classification of all plastic bottles.
In the Figure 5.23 there are shown incorrect detections from YOLOv3 and YOLOv3-Tiny models respectively. The Test 2 dataset was more challenging than Test 1 dataset. The biggest problem was in the detection and classification of plastic bags. We can see that YOLOv3 correctly detected menu boxes, but instead of detection and classification of plastic bag, plastic bottle was detected. YOLOv3-Tiny detected menu boxes but misclassified them with plastic bags. The present plastic bag was not detected.

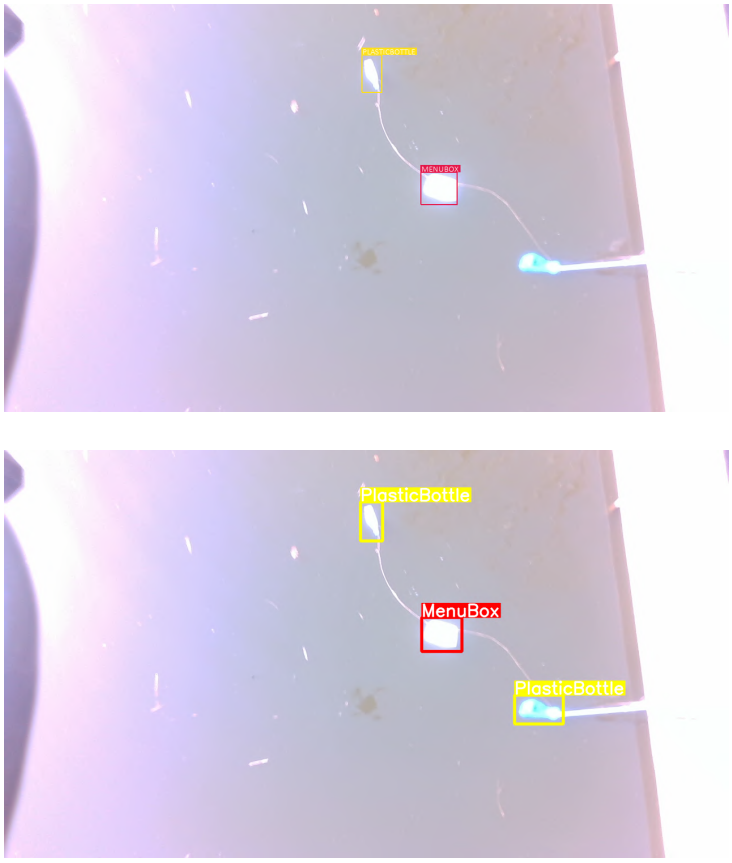


Figure 5.15: Comparison between SSD and YOLOv3 model.

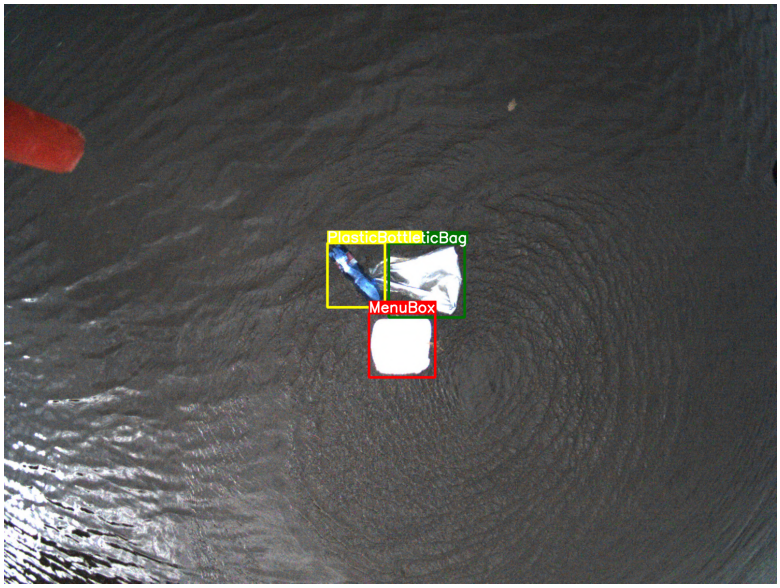


Figure 5.16: Detection by YOLOv3 model.

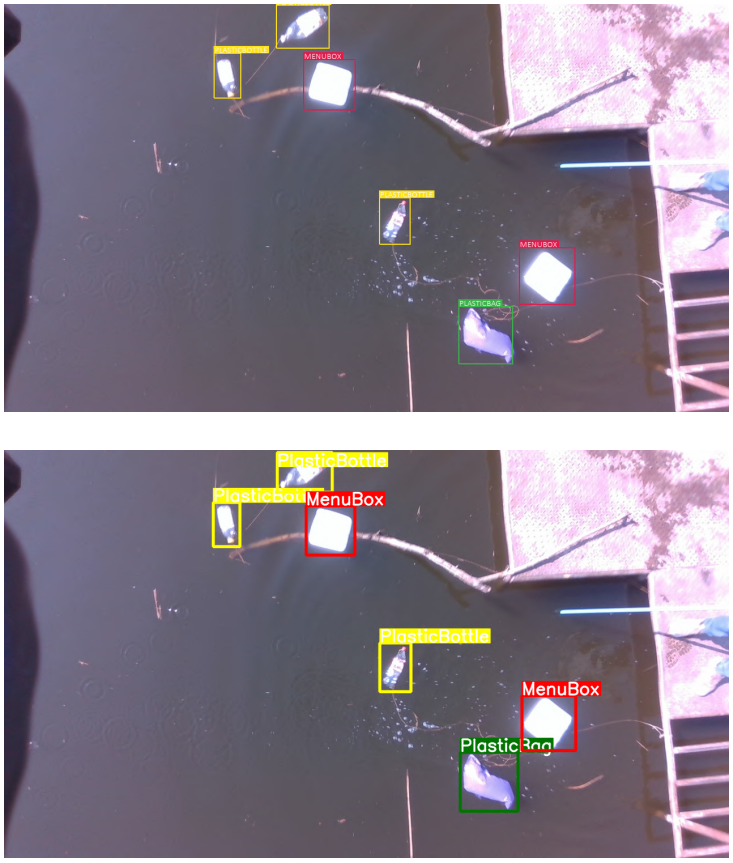


Figure 5.17: Correct detections by SSD and YOLOv3 models.

Figure 5.18: Correct detections by YOLOv3 model.

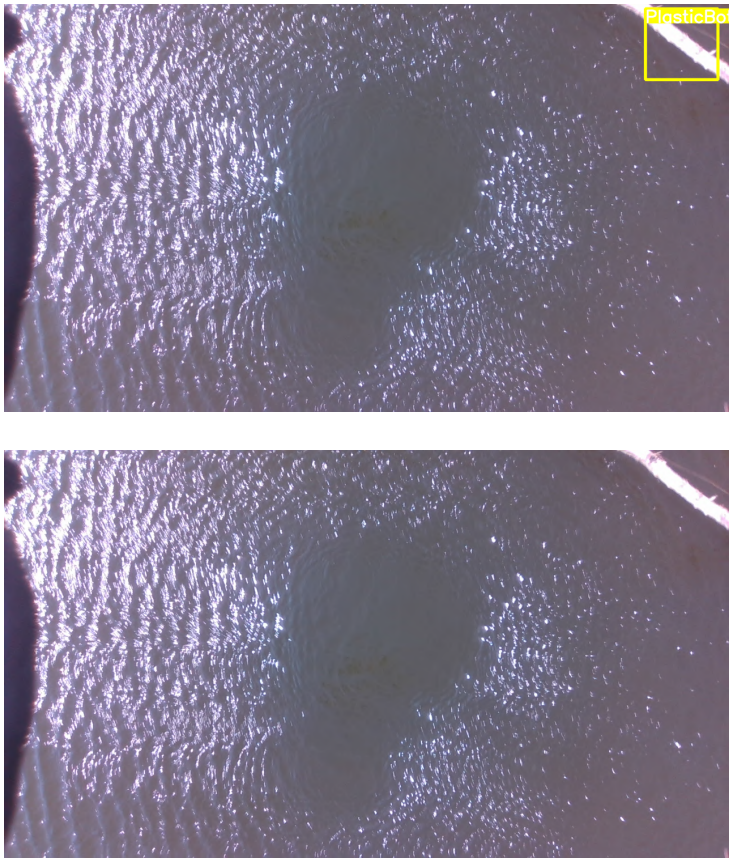


Figure 5.19: Comparison between YOLOv3-Tiny and YOLOv3 model.

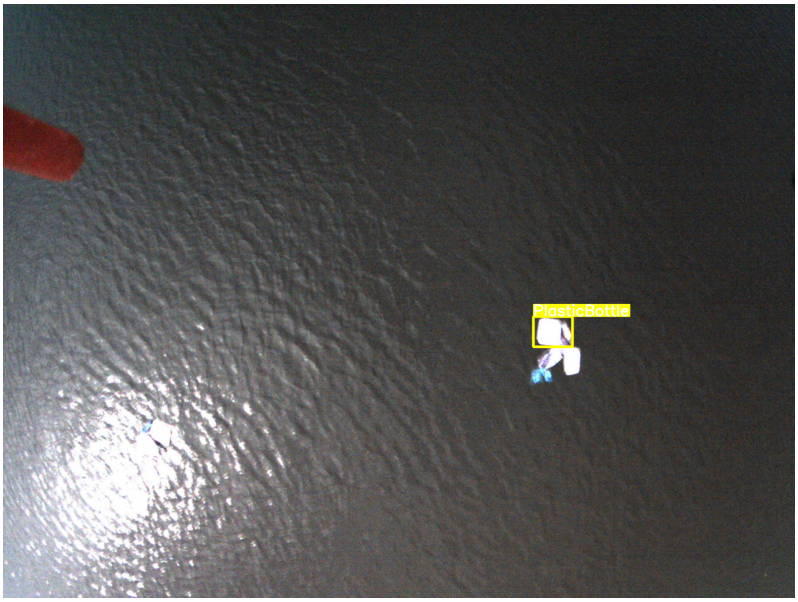


Figure 5.20: Failed detection by YOLOv3 model in image captured from $\sim 7m$.

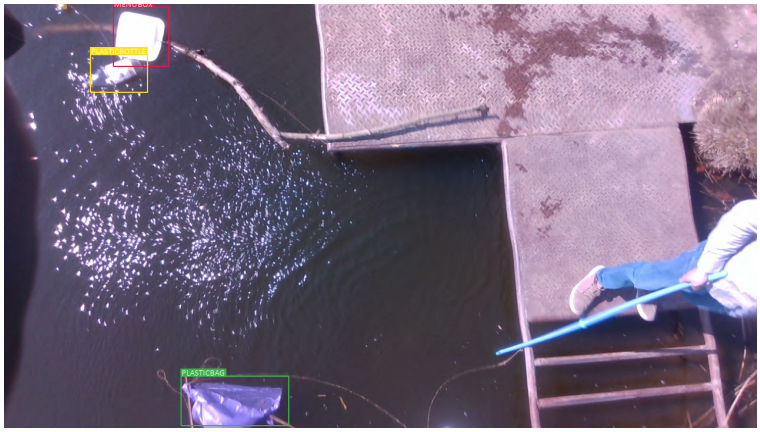


Figure 5.21: Correct detections by SSD model.

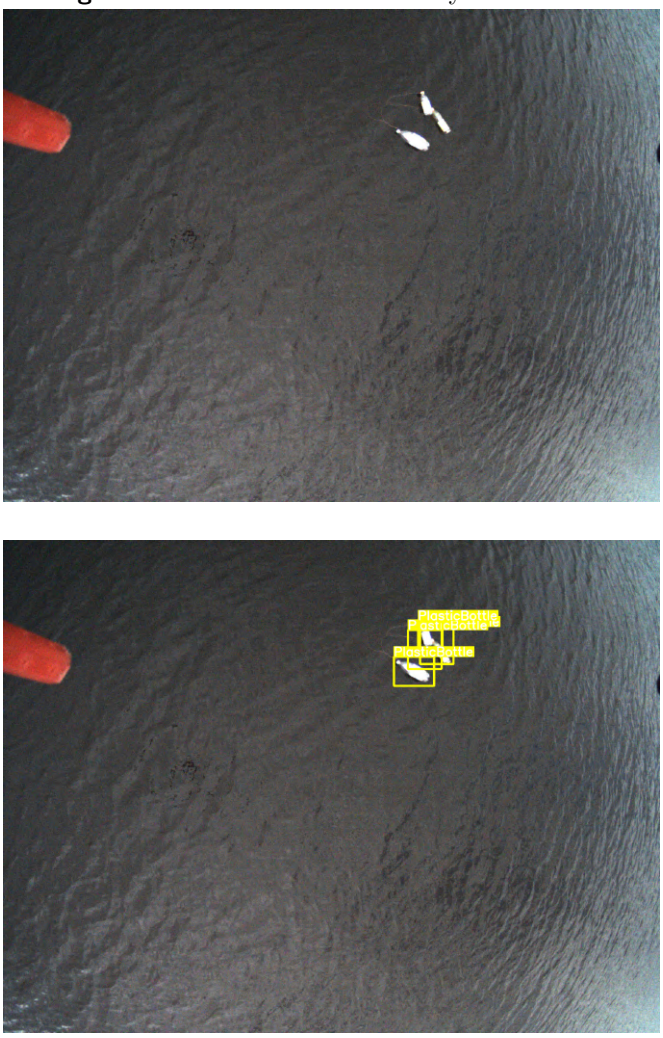


Figure 5.22: Comparison between SSD and YOLOv3-Tiny model.

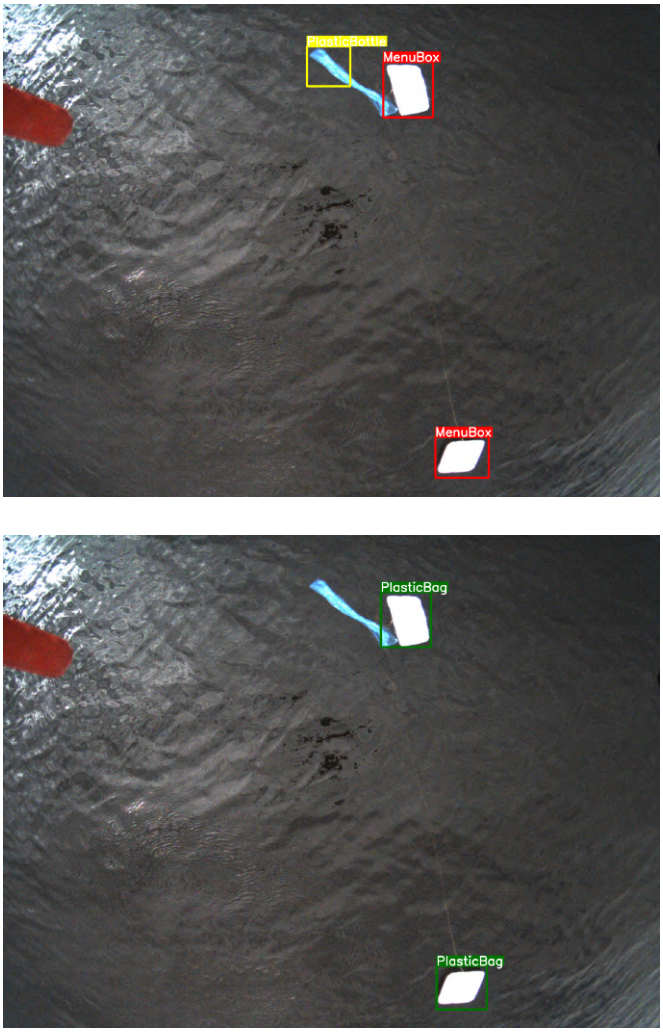


Figure 5.23: Comparison between YOLOv3 and YOLOv3-Tiny model.

Since the SORT tracking algorithm is based on tracking by detection paradigm, its performance is highly dependent on detection model performance. The highest multi-object tracking accuracy (MOTA) was achieved with detections provided by the YOLOv3 model on both test videos since the model achieved the highest mAP among all experimented models. The lowest count of Frags and ID switches was also observed with the YOLOv3 model. The second best performance was achieved with detections provided by the YOLOv3-Tiny model. In the performance measurement, it was shown that SORT does not add more computational load on hardware.

Our small experiments showed that lidar can sense the floating debris on the water surface. Lidar can provide complementary information to the task of detection of the floating debris, where the detection models fail. Noise in lidar data is observed when the rotors from UAV create small waves.

Chapter 6

Discussion

In this chapter we will discuss our goal of this thesis and propose the combination of detection and tracking model, which can be implemented on board of the UAV.
In this thesis we wanted to develop a detection-tracking system that can be implemented on board of the UAV for detection, classification and tracking of floating debris on the water surface, which can run in real-time. The detection and classification part of the proposed system is based on a deep learning model. The deep learning based model was used because it can learn and construct features of complex environments autonomously. We chose to classify three classes of most common floating debris, which are menu boxes, plastic bags and plastic bottles.
Main goal of choosing the tracking method was the abiltity to track multiple objects on the water surface and to not add computational load on top of an already computational demanding detection model.

We experimented with three detection models, which were SSD, YOLOv3 and YOLOv3-Tiny. Models were evaluated on 2 test datasets. Test 2 dataset was considered as more challenging as mentioned in chapter 4.
We experimented with hyperparameters and data augmentations during the training phase of detection models. Evaluation of the detection models after training shows, that the trained models can successfully detect and classify floating debris on the water surface. Experiments with choosing different thresholds of confidence score and IoU were gathered. These experiments were gathered because we wanted to know, how the models are confident and precise with their detections after training. If the models cannot provide sufficient detections of floating debris and we will send the UAV on unprecise location, this will be a waste of energy and the flight time of the UAV will

decrease.

SORT was proposed to be used for the multi-object tracking algorithm. Since the proposed method is based on tracking by detection paradigm, detections from detection models are needed on its input. The proposed algorithm was able to successfully track multiple objects on water surface. The performance of the SORT is dependent on the performance of the detection models. Methods based on machine or deep learning can add computational load on hardware. The proposed method can be suitable for implementation on detection models, since the proposed method is not based on either of them.

Results from the evaluation of detection models on test datasets show that the YOLOv3 achieved the highest mAP on both test datasets. Achieved mAP was 96.4% and 70.2% for Test 1 and Test 2 dataset respectively with confidence and IoU thresholds set both to 50%. SSD model achieved the lowest mAP on both of the test datasets. If we wanted to use the confidence threshold of 75%, YOLO models will be suitable. The drop of the mAP was in case of YOLOv3 model $\sim$ 6% and in the case of YOLOv3-Tiny $\sim$ 3% in the challenging dataset. We observed a significant drop in performance of the SSD model on Test 2 dataset. This drop shows that the model was not robust enough to perform well on the new data in comparison with the YOLO models.
Another important property that we evaluated, was the speed of the model with and without implemented tracking algorithm. The fastest performing model in both test datasets was the YOLOv3-Tiny model, which achieved 76 FPS and 63 FPS. The drop in speed was caused by the higher resolution of images in Test 2 dataset. YOLOv3 achieved almost real-time performance.

In the evaluation of the SORT tracking algorithm implemented on top of the detection models, achieved highest MOTA and MOTP was with the YOLOv3 combination, since SORT is highly dependent on the performance of the detection model. The lowest number of identity switches and highest number of MT targets were also obtained with the YOLOv3 used as the detection model. SORT showed up to be suitable for implementation on board of the UAV, because after implementation to the detection-tracking pipeline FPS remained unchanged.

We propose to use a combination of YOLOv3 and SORT as a tracking algorithm in detection-tracking system. YOLOv3 achieved the highest mAP among all proposed detection and classification models and speed in terms of FPS was almost real-time. With the sufficient hardware [33] on the board of the UAV it can run in real-time. If the UAV hardware will not be sufficient, we propose to use a lighter version of YOLOv3, which is YOLOv3-Tiny. The

SORT tracking algorithm is able to track floating debris on the water surface and will not add computational load and performance will improve with a better detection model.

Small experiments with the lidar showed its ability to sense the floating debris on the water surface and to provide complementary information to the detection task. Lidar was able to detect objects, where the detection models based on deep learning failed. Noise in data was observed when the UAV had low flight altitude and rotors created small waves on the water surface.

Chapter 7

Conclusion and future work

Our thesis dealt with the problem of detection and tracking floating debris on the water surface. The main goal of this thesis was to propose and implement a detection-tracking system, which can be able to detect, classify and track floating debris on the water surface. We wanted to classify three classes of most common marine debris, which were menu boxes, plastic bags and plastic bottles.

We tackled the problem of detection and classification with methods based on deep learning. These methods were used, because they can automatically learn to extract useful features from complex tasks. Deep learning methods need to have a lot of data on their input. Since we have not found any dataset with the UAV images of floating debris, we created our own custom. Dataset consisted of data collected from pond and river with different properties and challenges, which we described in chapter 4. Collected images were selected and annotated.
Three detection models based on deep learning were proposed. These models were SSD, YOLOv3 and YOLOv3-Tiny. Proposed models were evaluated with different confidence and IoU thresholds on two test datasets. Test 2 dataset was considered as more challenging. YOLOv3 achieved the highest mAP 96.4% and 70.2% on Test 1 and Test 2 dataset respectively, among all models with confidence and Iou thresholds set both to 50%. YOLOv3 also achieved almost real-time performance with 28 FPS and 22 FPS on Test 1 and Test 2 dataset. The fastest performing detection model with 76 FPS and 63 FPS on Test 1 and Test 2 dataset was YOLOv3-Tiny, which achieved the second highest mAP on both test datasets.

Tracking was tackled with the SORT tracking algorithm. Method can track multiple objects on the water surface and is based on the tracking by detection paradigm. Inputs to the algorithm are detections, which consist of bounding box coordinates, score and class from the detection model. We created our own annotated dataset for evaluation, which is described in chapter 4. The evaluation showed that the algorithm is highly dependent on the performance of the detection model. SORT performed well with the YOLOv3 model achieving MOTA of 85.1% and 52.7% on Test 1 and Test 2 video. The highest number of mostly tracked objects with the lowest number of identity switches was achieved. Evaluation of speed of the system showed that SORT does not add computational load to already computational demanding detection model based on deep learning.

We proposed to use a combination of YOLOv3 and SORT as a tracking algorithm to be implemented on UAV with sufficient hardware. This combination achieved the highest results in our evaluations and can run almost in real-time. When the hardware will not be sufficient, implementation of YOLOv3-Tiny with SORT is suggested.

Since the main sensor for the detection-tracking system in our assignment was a camera, small experiments with lidar were gathered. Experiments showed potential usage of the lidar providing complementary information to the detection task. Lidar was able to detect objects, where the proposed detection models failed.

Our thesis presents the first step in the new area of research in the Multi-robot Systems team at CTU in Prague, focused on the complex system of removing floating debris from water surface by UAVs. The most important part of the future work will be to implement and test the proposed system on the real UAV.
Collection of new data from different localities will be needed for the improvement of detection models performance. The detection task can be improved with the creation of sensor fusion with the lidar data.

Bibliography

[1] A. McIlgorm, H. F. Campbell, and M. J. Rule, "Understanding the economic benefits and costs of controlling marine debris in the APEC region"," *A report to the Asia-Pacific Economic Cooperation Marine Resource Conservation Working Group by the National Marine Science Centre (University of New England and Southern Cross University)*, pp. 1–95, Dec. 2008.

[2] K. Engin, M. Tran, R. Connor, *et al.*, "The united nations world water development report 2018: nature-based solutions for water," *UNESCO*, Mar. 2018.

[3] S. N. Hasany, S. S. Zaidi, S. A. Sohail, *et al.*, "An autonomous robotic system for collecting garbage over small water bodies," *2021 6th International Conference on Automation, Control and Robotics Engineering (CACRE)*, pp. 81–86, 2020.

[4] J. Watanabe, Y. Shao, and N. Miura, "Underwater and airborne monitoring of marine ecosystems and debris," *Journal of Applied Remote Sensing*, vol. 13, p. 17091–17099, Oct. 2019.

[5] K. Yun, L. Nguyen, T. Nguyen, *et al.*, "Small target detection for search and rescue operations using distributed deep learning and synthetic data generation," *Pattern Recognition and Tracking XXX*, vol. 10995, pp. 1099507:1–6, Apr. 2019.

[6] P. R. Pawar, S. S. Shirgaonkar, and R. B. Patil, "Plastic marine debris: Sources, distribution and impacts on coastal and ocean biodiversity," *PENCIL Publication of Biological Sciences (OCEANOGRAPHY)*, vol. 3, pp. 40–54, Jan. 2016.

[7] M. Tharani, A. W. Amin, M. Maaz, *et al.*, "Attention neural network for trash detection on water channels," *arXiv preprint arXiv:2007.04639*, pp. 1–6, July 2020.

[8] C. van Lieshout, K. van Oeveren, T. van Emmerik, *et al.*, "Automated River Plastic Monitoring Using Deep Learning and Cameras," *Earth and Space Science*, vol. 7, pp. 1–14, Aug. 2020.

[9] F. F. Putra and Y. D. Prabowo, "Low resource deep learning to detect waste intensity in the river flow," *Bulletin of Electrical Engineering and Informatics*, vol. 10, pp. 2724–2732, Oct. 2021.

[10] N. A. Zailan, A. S. M. Khairuddin, U. Khairuddin, *et al.*, "Yolo-based Network Fusion for Riverine Floating Debris Monitoring System," *2021 International Conference on Electrical, Communication, and Computer Engineering (ICECCE)*, pp. 1–5, June 2021.

[11] K. Kylili, I. Kyriakides, A. Artusi, *et al.*, "Identifying floating plastic marine debris using a deep learning approach," *Environmental Science and Pollution Research*, p. 17091–17099, June 2019.

[12] W. Zhang, X. Gao, C. Yang, *et al.*, "A object detection and tracking method for security in intelligence of unmanned surface vehicles," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–13, Oct. 2020.

[13] X. Li, M. Tian, S. Kong, *et al.*, "A modified YOLOv3 detection method for vision-based water surface garbage capture robot," *International Journal of Advanced Robotic Systems*, vol. 17, pp. 172988142093271:1–11, May 2020.

[14] V. A. Feraru, R. E. Andersen, and E. Boukas, "Towards an Autonomous UAV-based System to Assist Search and Rescue Operations in Man Overboard Incidents," *2020 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, pp. 57–64, Nov. 2020.

[15] L. Qingqing, J. Taipalmaa, J. P. Queralta, *et al.*, "Towards Active Vision with UAVs in Marine Search and Rescue: Analyzing Human Detection at Variable Altitudes," *2020 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, pp. 65–70, Nov. 2020.

[16] G. Niu, J. Li, S. Guo, *et al.*, "SuperDock: A Deep Learning-Based Automated Floating Trash Monitoring System," *2019 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, p. 1035–1040, Dec. 2019.

[17] B. Zhang, X. Qian, R. Yang, *et al.*, "Water Surface Target Detection Based on Improved YOLOv3 in UAV Images," *Association for Computing Machinery*, pp. 47–53, Feb. 2021.

[18] D. Duarte, M. Pereira, and A. Pinto, "Multiple Vessel Detection and Tracking in Harsh Maritime Environments," *OCEANS 2021: San Diego – Porto*, pp. 1–5, 2021.

[19] Y. Shan, X. Zhou, S. Liu, *et al.*, "SiamFPN: A Deep Learning Method for Accurate and Real-Time Maritime Ship Tracking," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, pp. 315–325, 2021.

[20] X. Chen, X. Xu, Y. Yang, *et al.*, "Augmented Ship Tracking Under Occlusion Conditions From Maritime Surveillance Videos," *IEEE Access*, vol. 8, pp. 42884–42897, 2020.

[21] D. Qiao, G. Liu, J. Zhang, *et al.*, "M3C: Multimodel-and-Multicue-Based Tracking by Detection of Surrounding Vessels in Maritime Environment for USV," *IEEE Access*, vol. 8, pp. 1–18, June 2019.

[22] H. Liao, W. Wang, S. Wang, *et al.*, "Multi-Scale Ship Tracking Based On Maritime Monitoring Platform," *2020 15th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, pp. 945–949, 2020.

[23] O. T. Arnegaard, F. S. Leira, H. H. Helgesen, *et al.*, "Detection of objects on the ocean surface from a UAV with visual and thermal cameras: A machine learning approach," *2021 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 81–90, 2021.

[24] A. Panico, L. Z. Fragonara, and S. Al-Rubaye, "Adaptive Detection Tracking System for Autonomous UAV Maritime Patrolling," *2020 IEEE 7th International Workshop on Metrology for AeroSpace (MetroAeroSpace)*, pp. 539–544, 2020.

[25] S. Kapania, D. Saini, S. Goyal, *et al.*, "Multi Object Tracking with UAVs using Deep SORT and YOLOv3 RetinaNet Detection Framework," *Proceedings of the 1st ACM Workshop on Autonomous and Intelligent Mobile Systems*, pp. 1–6, Jan. 2020.

[26] P. Nousi, I. Mademlis, I. Karakostas, *et al.*, "Embedded UAV Real-Time Visual Object Detection and Tracking," *2019 IEEE International Conference on Real-time Computing and Robotics (RCAR)*, pp. 708–713, Aug. 2019.

[27] S. Xu, A. Savvaris, S. He, *et al.*, "Real-time Implementation of YOLO+JPDA for Small Scale UAV Multiple Object Tracking," *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 1336–1341, June 2018.

[28] Y. Liu, Q. Wang, H. Hu, *et al.*, "A Novel Real-Time Moving Target Tracking and Path Planning System for a Quadrotor UAV in Unknown Unstructured Outdoor Scenes," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, pp. 2362–2372, Nov. 2019.

[29] I. Saetchnikov, V. Skakun, and E. Tcherniavskaia, "Efficient objects tracking from an unmanned aerial vehicle," *2021 IEEE 8th International Workshop on Metrology for AeroSpace (MetroAeroSpace)*, pp. 221–225, 2021.

[30] H. Shen, D. Lin, and T. Song, "A real-time siamese tracker deployed on UAVs," *Journal of Real-Time Image Processing*, vol. 19, p. 463–473, Jan. 2022.

[31] W. Li, J. Mu, and G. Liu, "Multiple Object Tracking with Motion and Appearance Cues," *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pp. 161–169, Aug. 2019.

[32] D. Biswas, H. Su, C. Wang, *et al.*, "Speed Estimation of Multiple Moving Objects from a Moving UAV Platform," *ISPRS International Journal of Geo-Information*, vol. 8, pp. 1–15, May 2019.

[33] S. Hossain and D. j. Lee, "Deep Learning-Based Real-Time Multiple-Object Detection and Tracking from Aerial Imagery via a Flying Robot with GPU-Based Embedded Devices," *Sensors*, vol. 19, pp. 1–24, July 2019.

[34] C. Lusardi, A. M. N. Taufique, and A. Savakis, "Robust Multi-Object Tracking Using Re-Identification Features and Graph Convolutional Networks," *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pp. 3861–3870, 2021.

[35] X. Wu, W. Li, D. Hong, *et al.*, "Deep Learning for uav-based Object Detection and Tracking: A Survey," *IEEE Geosci. Remote Sens. Mag.,*, pp. 1–24, 2021.

[36] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a Convolutional Neural Network," *2017 International Conference on Engineering and Technology (ICET)*, pp. 1–6, 2017.

[37] "Introduction to convolutional neural network." `https://medium.com`. online, February 2021.

[38] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," *arXiv preprint arXiv:1804.02767*, pp. 1–6, Apr. 2018.

[39] "All you need to know about yolo v3 (you only look once)." `https://dev.to`. online, February 2021.

[40] "Intersection over union (iou) for object detection." `https://www.pyimagesearch.com`. online, February 2021.

[41] "Yolov3 implemented by ultralytics." `https://github.com/`. online, April 2021.

[42] W. Liu, D. Anguelov, D. Erhan, *et al.*, "SSD: Single shot multibox detector," *ECCV*, pp. 1–17, Dec. 2016.

[43] K. Simonyan and A. Zisserman, "VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION," *ICLR 2015*, pp. 1–14, Apr. 2015.

[44] "Understanding ssd multibox—real-time object detection in deep learning." `https://towardsdatascience.com`. online, February 2021.

[45] "A-pytorch-tutorial-to-object-detection." `https://github.com`. online, February 2021.

[46] "Confusion Matrix for Your Multi-Class Machine Learning Model." `https://towardsdatascience.com/`. online, April 2021.

[47] "object_detection_confusion_matrix." `https://github.com/`. online, April 2021.

[48] A. Bewley, Z. Ge, L. Ott, *et al.*, "Simple online and realtime tracking," *2016 IEEE International Conference on Image Processing (ICIP)*, pp. 3464–3468, 2016.

[49] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," *2017 IEEE International Conference on Image Processing (ICIP)*, pp. 3645–3649, 2017.

[50] "SORT - A simple online and realtime tracking algorithm." `https://github.com/abewley/sort`. online, April 2021.

[51] J. Luiten and A. Hoffhues, "TrackEval." `https://github.com/JonathonLuiten/TrackEvalt`. online, April 2021.

[52] "Labelimg." `https://github.com/tzutalin/labelImg`. online, February 2021.

[53] "FFmpeg tool." `http://ffmpeg.org/`. online, April 2021.

[54] "Computer Vision Annotation Tool (CVAT)." `https://github.com/openvinotoolkit/cvat`. online, April 2021.

[55] F. Zhuang, Z. Qi, K. Duan, *et al.*, "A Comprehensive Survey on Transfer Learning," *Proceedings of the IEEE*, vol. 109, pp. 43–76, 2021.

[56] "Breaking down mean average precision (map)." `https://towardsdatascience.com/`. online, April 2021.

[57] K. Bernardin and R. Stiefelhagen, "Evaluating Multiple Object Tracking Performance: The CLEAR MOT Metrics," *EURASIP Journal on Image and Video Processing*, vol. 2008, pp. 1–10, May 2008.

[58] M. W. M. Said, "LiDAR and Vision Based Pack Ice Field Estimation for Aided Ship Navigation," *PROJECT REPORT*, pp. 1–18, Aug. 2017.

Appendix A

CD content

Thesis - folder contains the diploma thesis in pdf file
Evaluation_Tracker - tool for evaluation of the tracking algorithm SORT
SSD_pytorch - codes for the training, testing and inference of the SSD model
YoloV3_pytorch - codes for the training, testing and inference of the YOLO models

Table A.1: Directories on the CD.

MASTER‘S THESIS ASSIGNMENT

I. Personal and study details

Student's name: **Ukleh Adam** Personal ID number: **453060**

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

Department / Institute: **Department of Cybernetics**

Study program: **Cybernetics and Robotics**

Branch of study: **Cybernetics and Robotics**

II. Master’s thesis details

Master’s thesis title in English:

Detection and Tracking of Objects on Water Surface

Master’s thesis title in Czech:

Detekce a sledování objektů na vodní hladině

Guidelines:

Goal of this thesis is to propose, implement and verify an approach for detection, classification and tracking of floating objects on water surface using an onboard UAV camera. The main motivation is designing a perception system for autonomous removing debris from polluted water surfaces by aerial robots. Following tasks will be solved:
1. Review and learn methods for object detection, classification and tracking on water surface.
2. Design and implement a method for detection and classification of floating objects on water surface.
3. Design and implement a method for tracking of objects on water surface.
4. Perform verification experiments with the proposed methods in real-world conditions (data for the object detection and tracking will be gained by a real UAV flying above real objects floating on water surface) and statistically evaluate their performance.

Bibliography / sources:

[1] B. Zhang, X. Qian, R. Yang, et al., Water Surface Target Detection Based on Improved YOLOv3 in UAV images, Association for Computing Machinery 2021
[2] V. A. Feraru, R. E. Andersen, and E. Boukas, Towards an Autonomous UAV-based System to Assist Search and Rescue Operations in Man Overboard Incidents, 2020 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)
[3] Yiming Li, Changhong Fu, Ziyuan Huang, et al., Keyfilter-Aware Real-Time UAV Object Tracking, 2020 IEEE International Conference on Robotics and Automation (ICRA)
[4] T. Ba c a, M. Petrlı k, M. Vrba, et al. , The MRS UAV System: Pushing the frontiers of reproducible research, real-world deployment, and education with autonomous unmanned aerial vehicles, https://arxiv.org/abs/2008.08050, 2020.

Name and workplace of master’s thesis supervisor:

doc. Ing. Martin Saska, Dr. rer. nat. Multi-robot Systems FEE

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

Date of master’s thesis assignment: **31.01.2022** Deadline for master's thesis submission: **20.05.2022**

Assignment valid until: **30.09.2023**

doc. Ing. Martin Saska, Dr. rer. nat. Supervisor’s signature	prof. Ing. Tomáš Svoboda, Ph.D. Head of department’s signature	prof. Mgr. Petr Páta, Ph.D. Dean’s signature

III. Assignment receipt

The student acknowledges that the master's thesis is an individual work. The student must produce his 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