Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Engineering Cybernetics

Herman Kulild Dragesund

Obstacle detection and avoidance system for USV FishOtter

Master's thesis in Cybernetics and Robotics
Supervisor: Jo Arve Alfredsen
Co-supervisor: Nikolai Lauvås

February 2022

NTNU
Norwegian University of
Science and Technology

Herman Kulild Dragesund

Obstacle detection and avoidance system for USV FishOtter

Master's thesis in Cybernetics and Robotics
Supervisor: Jo Arve Alfredsen
Co-supervisor: Nikolai Lauvås
February 2022

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Engineering Cybernetics

NTNU
Norwegian University of
Science and Technology

Abstract

This thesis aims to develop a collision avoidance algorithm for a small Unmanned Surface Vehicle (USV) used in the Robotic Fish Tracking project at the Norwegian University of Science and Technology. The algorithm uses a low cost camera sensor which is chosen for this project and is mounted on the USV.

A literature review of object detection algorithms, as well as georeferencing methods using camera sensors, is conducted. Based on this review a georeferencing method is chosen and modified for the specific purpose of this project. The method chosen is based on the work of Helgesen et al. (2020). It utilizes a single stage object detection algorithm called YOLO which detects and locates objects in the image. Based on the objects' position in the image and the camera's elevation, the distance between the USV and the detected objects are estimated.

A data set containing a video stream and GPS coordinates is recorded in the Trondheims fjord using the USV. The video stream contains images of boats, docs and buoys which are intended to be detected by the algorithm presented in this thesis. The GPS data contain coordinates of the USV's position and the relevant objects. The data set is later used to test the accuracy of the algorithm on an external computer. It is concluded that the accuracy is too low to function as a collision avoidance system at the current stage. However, it shows some promise in that it is able to detect the majority of the relevant objects and provides a ”ballpark” estimation of the distance to these objects. In future work it can be used as a starting point for further development of an algorithm which avoids collisions during operation.

Sammendrag

Denne avhandlingen har som målsetning å utvikle og implementere en algoritme for kollisjonsunngåelse for et lite ubemannet marint fartøy (USV). Fartøyet brukes i et prosjekt ved Norges Teknisk-Naturvitenskapelige Universitet kalt ”Robotic Fish Tracking Project”, hvilket har som målsetning å bruke autonome farkoster til å spore marine ressurser. Algoritmen bruker en optisk sensor med lav kostnad, hvilket er valgt ut som en del av dette prosjektet.

Et litteraturstudie som ser på ulike algoritmer for objektdeteksjon og avstandsestimering er gjennomført. En metode for avstandsestimering er så valgt ut og modifisert for prosjektet. Den valgte metoden er en modifisert versjon av den som er beskrevet i Helgesen et al. (2020). Den bruker først en objektdeteksjonsalgoritme kalt YOLO til å detektere og lokalisere objekter i bildet. Deretter estimerer den avstanden til disse objektene basert på hvor de er plassert i bildet og høyden til kameraets plassering.

Et datasett med video og GPS-koordinater ble spilt inn med USVen i Trondheimsfjorden. Videostrømmen inneholder bilder av båter, brygger og bøyer, som det er tiltenkt at algoritmen, presentert i denne avhandlingen, skal detektere. GPS dataene inneholder posisjonen til USVen og de relevante gjenstandene. Datasettet er så brukt til å teste algoritmen med en ekstern datamaskin. Det konkluderes med at algoritmen på nåværende stadium ikke er tilstrekkelig nøyaktig til å bli brukt til kollisjonsunngåelse. Den viser likevel noe potensialet ved at den oppdager de fleste relevante gjenstandene, og klarer å gi et omtrentlig estimat av avstanden til dem. Arbeidet i denne avhandlingen kan derfor brukes som et utgangspunkt for videre arbeid som kan lede til et fullverdig system for kollisjonsunngåelse

Preface

This work is the result of the course TTK4900 - Engineering Cybernetics, Master's Thesis, at the Norwegian University of Science and Technology (NTNU) at the department of Engineering Cybernetics.

I would like to thank my supervisor Jo Arve Alfredsen and co-supervisor Nikolai Lauvås for their continued guidance and assistance during this project.

I also want to thank my fellow students Nikolai, Magne, Ole-Jørgen, Halvor, Gustav, Simen, Daniel, Sander, Aleksander and Rune. Their continued support and friendship during long days at campus have been much appreciated. The last 5 years would neither have been possible nor nearly as enjoyable without them.

I want to thank my flatmates at Berg Studenby: Pernille, Sindre, Anna, Hege, Tove, Simon and Linn for acting as my second family during my first years in Trondheim. The privilege of living with a group of friends who one can share life's ups and downs with has been greatly appreciated, especially when first arriving in Trondheim.

I'm also very grateful for the numerous adventures, and friendship with the people in the Marketing group at Studentersamfundet. My time in Trondheim has been much more memorable and enjoyable because of them.

Lastly I want to thank my family for their continued support since day one.

Table of Contents

List of Tables

List of Figures

Abbreviations

Either

USV	=	Unmanned Surface Vessel
PoE	=	Power over Ethernet
IR	=	Infra Red
CNN	=	Convolutional Neural Network
R-CNN	=	Regions with Convolutional Neural Networks
SSD	=	Single Shot Multibox Detector
YOLO	=	You Only Look Once
IOU	=	Intersection of Union
NTNU	=	Norwegian University of Science and Technology
RMSE	=	Root Mean Squared Error
IMU	=	Internal Measurement Unit

Chapter 1

Introduction

1.1 Background

Our ability to manage our marine environment in a productive and sustainable manner relies on our understanding of the behavior and distribution of marine living resources. A project at The Norwegian University of Science and Technology (NTNU) therefore aims at integrating autonomous vehicles and acoustic fish telemetry to localize and track migrating fish and other marine assets. This project is named ”Robotic fish tracking - integration of AUV/USV and acoustic fish telemetry” (Alfredsen).

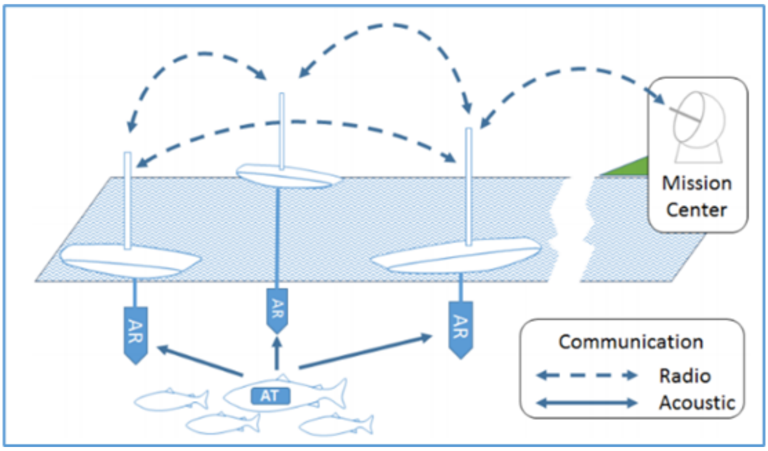


Figure 1.1: Consept visualization of Project - Robotic fish tracking - integration of AUV/USV and acoustic fish telemetry.

A key part of this project is the development of low cost autonomy for the USV, such that constant human intervention is not needed. For autonomous vehicles to function they need to use sensors to obtain a situational awareness of the vehicles' surroundings. Commonly used sensors for this application include thermal camera, electro-optical camera, stereo camera, radar and lidars (Campbell et al., 2018). Radar, lidars and thermal cameras have relatively high costs and are beyond the price range of this project. Fortunately, computer vision algorithms have made great progress in recent years and have proven to be effective at providing contextual information of the surroundings. Furthermore, the advent of object detection algorithms has enabled directional information of detected objects to be extracted from cameras. Monocular cameras do however not provide range information explicitly, but techniques for range estimation based on images joined with other known information do exist.

This thesis therefore aims at using a monocular camera to detect and classify objects in front of the USV, and estimate the position of the objects relative to its own. Ultimately, the goal for this is to serve as a basis to develop an collision avoidance algorithm in future work. With the deployment of such a system the USV could operate in a safe way without constant human surveillance.

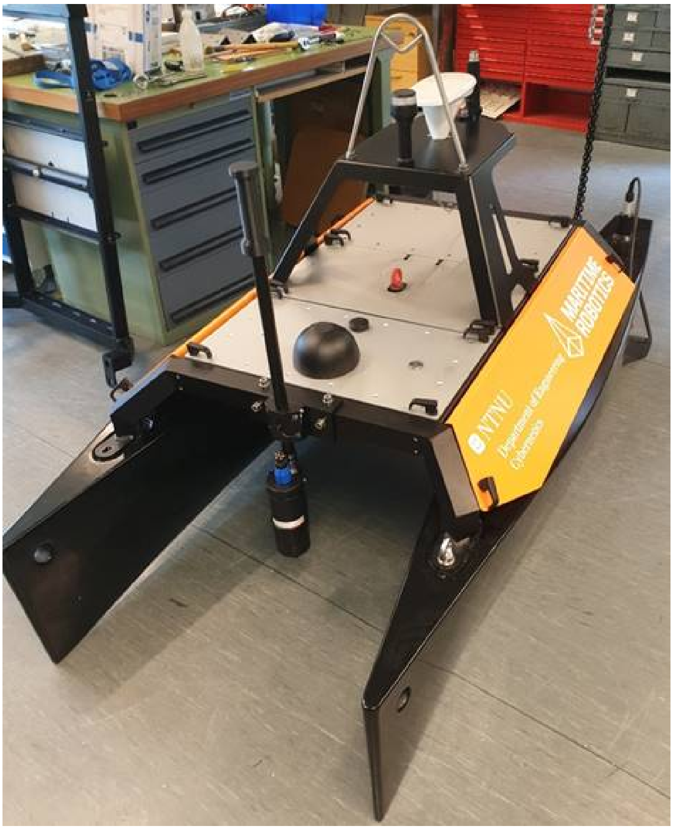


Figure 1.2: USV used in the project.

1.2 Thesis outline

The thesis is structured such that chapter 2 will present the relevant theory for the thesis. Chapter 3 will then describe some of the existing work, related to this thesis. In chapter 4 the methods and experiment conducted in the thesis will be explained, and the results are presented in chapter 5. Chapter 6 will then discuss these results, as well as possible improvements. Finally a conclusion and suggestions for future work is provided in chapter 7.

Chapter 2

Theory

This chapter covers some of the theory applied in this thesis. The first part explains the basic workings of artificial neural networks, which is the backbone of the object detection algorithm used in this work. Next, several object detection algorithms are presented. This is because making an informed choice of object detection algorithm was an essential part of this thesis. Finally the theory behind the applied range estimation method is presented.

2.1 Artificial Neural Networks

The goal of artificial neural networks is to produce a desired output based on an given input. They are highly effective for many applications which are not easily programmed explicitly. A common example of this is the task of classifying images, which will be further explained in this chapter. Artificial neural networks get their name because they are inspired by how the network of neurons in our brain works, though it is worth noting that they do not try to mimic the function of the brain. The networks consist of nodes which can receive a signal from another node, process it, and send it to its output nodes. A node processes the input through what's called an activation function, which defines the output based on the input signal. These nodes are structured in consecutive layers where the nodes in one layer receives signals from the nodes in the previous layer, and sends its output signals to the next layers. The first layer is called the input layer, the last is called the output layer and the ones in between are called hidden layers. A visualisation of this is provided in figure 2.1.

Traning Neural Networks

In order to make the neural network produce the desired output it needs to be trained. To train the network, a data set with input and the desired output must be provided. One example of this is an black and white image of a handwritten number from 0 to 9. The input nodes can then pass the brightness of a pixel to the hidden layers, and the output node would be the possible numbers from 0 to 9. If a drawing of the number 3 was passed,

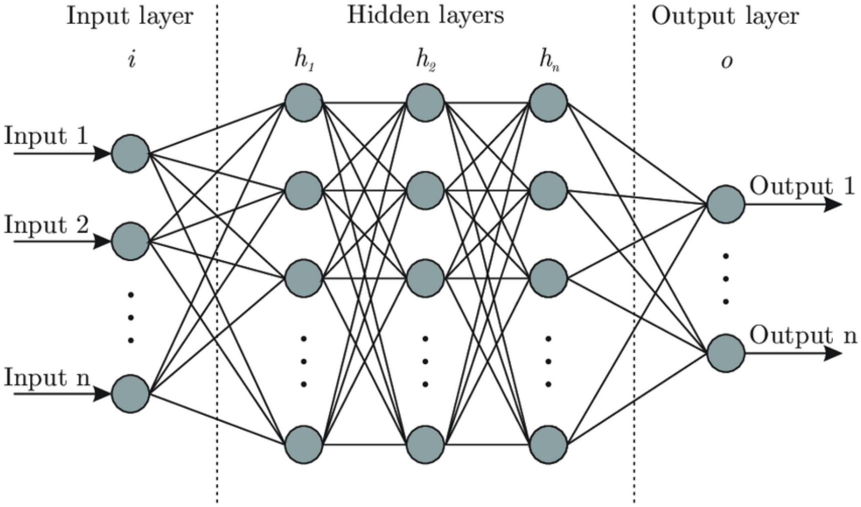


Figure 2.1: Visualisation of a generic neural network source: Bre et al. (2017)

the desirable output would be 1 at the output node representing 3, and 0 for the remaining output nodes.

A cost function can then be calculated based on the difference between the desired output and the actual output. We wish to tune the parameters in the activation function to minimise this cost function. To do this the method of gradient descent is applied. In short this method calculates the gradient from the cost function, and employs this to tune the parameter to reduce the loss (Kwiatkowski, 2021). When repeated multiple times and trained on thousands of images, the network is able to produce a valuable result.

Convolutional Neural Networks

Convolutional Neural Networks (CNN) is a type of Artificial Neural Network which is well suited for object detection and classification of images. It consists of nodes with activation functions structured in several hidden layers. Some of these layers are convolutional layers, which are particularly effective at detecting patterns in images (Albawi et al., 2017). Convolutional layers consist of filters, which usually consist of a kernel being a 3x3 matrix, which are convoluted with the image. Different types of filters will be able to detect different types of shapes. For example, a filter being:

$$\begin{bmatrix} -1 & -1 & -1 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix} \tag{2.1}$$

Will amplify horizontal edges. While a filter like:

$$\begin{bmatrix} -1 & 1 & 0 \\ -1 & 1 & 0 \\ -1 & 1 & 0 \end{bmatrix} \tag{2.2}$$

will amplify vertical edges.

Usually a CNN is structured such that the first convolutional layers detect basic structures like edges and corners, while the later layers pick up more complex structures. It is also common to use pooling layers in CNN. These layers are responsible for reducing the spatial size of convoluted features, hence decreasing the computational power necessary to run the network. This is crucial because CNNs can be quite computationally demanding. Furthermore, it allows for more high level structures, which are rotational and positional invariant, to be extracted. Pooling layers can also help prevent overfitting, which is when a network learns too specific features from the training set instead of generalising from the set. This causes the network to perform well on the training set but badly on the test set. There are two common types of pooling layers: Max Pooling and Average Pooling. Max Pooling returns the maximum value from the area covered by the kernel, while average pooling returns the average value. See figure 2.2 for visualisation.

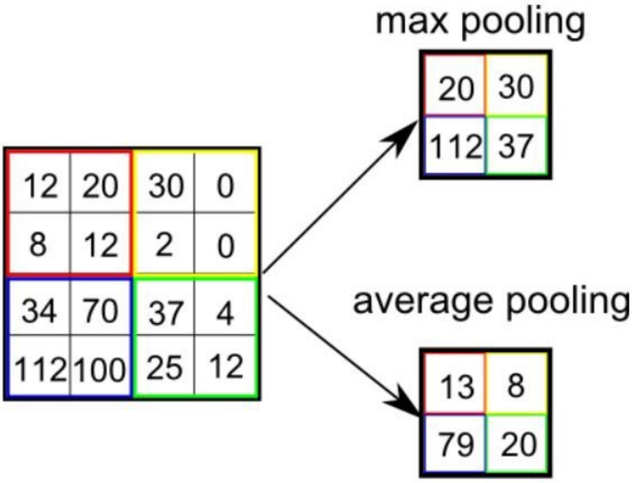


Figure 2.2: Visualisation of max pooling and average pooling. Source: Saha (2018)

2.2 Object detection

.

Object detection is the process of identifying and locating objects within an image or video. This is a considerably more challenging process than image classifying. In image classification the image is only labelled, such that an image containing three cats will be labelled cat. In object detection the problem is twofold, the image is labeled cat and bounding boxes are drawn around each cat, displaying where in the image the cats are. See figure 2.3 for visualisation.

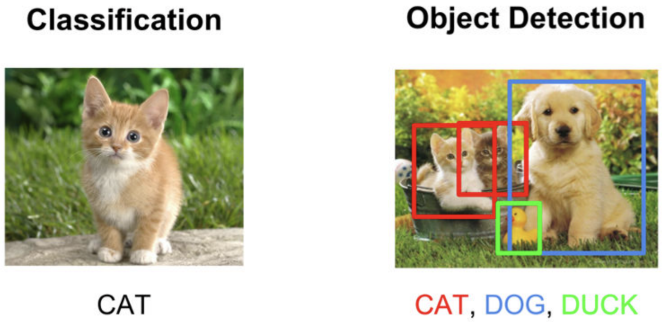


Figure 2.3: Visualisation of the difference between classification and object detection in images. Source: Hulstaert (2018)

2.2.1 Sliding Window

A sliding window approach is the most basic and straightforward way to solve the object detection problem. It works by choosing a sliding window which looks at a small segment of the image, usually starting in the top left corner, and runs a CNN on this segment. The CNN will return a prediction of possible objects in this segment and if objects are detected with high confidence, a bounding box is drawn around it. The sliding window is then moved a given number of pixels to the right and the process repeats. This is repeated until the whole image is covered. Because objects have different sizes in the image, depending on the objects size in the real world and the distance from the camera, sliding windows of different sizes needs to be tried and hence the whole process needs to be repeated several times. This makes the algorithm very computationally expensive and not suited for real time applications. Furthermore, the accuracy of the bounding boxes heavily depend on the number of different sized sliding windows applied and are therefore usually quite inaccurate. In 2013 the ImageNet Large Scale Visual Recognition Challenge was won by applying this technique (Sermanet et al., 2013), but in later years there have been several improvements leading to shorter run times and higher accuracy.

2.2.2 Regions with Convolutional Neural Networks

Regions with Convolutional Neural Networks (R-CNN) (Girshick et al., 2013) use a slightly different approach. It groups the image into regions based on texture, colour and intensity. These regions are then passed to a CNN, which makes classification predictions for each region, and if the confidence level is sufficiently high the region's bounding box is returned with the predicted classification. The R-CNN algorithm has been improved on several times with the Fast R-CNN (Girshick, 2015) and Faster R-CNN (Ren et al., 2017). However, faster and more accurate algorithms have since been developed.

2.2.3 YOLO

YOLO (You Only Look Once) is a state of the art object detection algorithm, and is a single stage detector, meaning it does all the computational work in one network (Redmon et al., 2016; Redmon and Farhadi, 2017, 2018). As the name implies it only looks once at the image, in contrast to Sliding window and R-CNN which looks several times at segments of the image. The YOLO algorithm solves this by dividing the image into a $S \times S$ sized grid. Each cell in the grid then predicts bounding boxes with a confidence score for each bounding box. A cell is responsible for detecting those objects which have their centre inside that cell. The confidence score is defined as:

$$P(Object) \times IOU_{pred}^{truth} \tag{2.3}$$

and reflects how confident the model is that the box contains an object and how accurate the size, shape and location of the box is. IOU is the Intersection of Union between the predicted bounding box and the ground truth. It is defined as the area of overlap divided by the area of union, see figure 2.4 for visualisation.

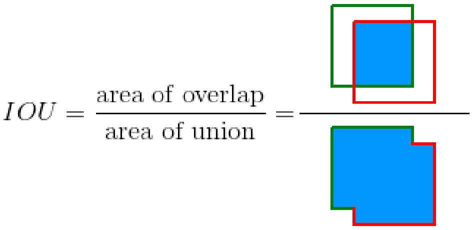


Figure 2.4: Visualisation of Intersection of Union. Source: Padilla et al. (2020)

When predicting the bounding boxes, dimension clusters are used as anchor boxes. 4 coordinates are predicted for each box: t_x, t_y, t_w and t_h, corresponding to the x-coordinate, y-coordinate, width and height of the bounding box respectively. Additionally, each cell predicts C conditional class probabilities:

$$P_r(Class_i|Object) \tag{2.4}$$

To acquire the class-specific confidence scores for each bounding box, this is multiplied with the confidence score, resulting in:

$$P_r(Class|Object) \times P(Object) \times IOU_{pred}^{truth} = P_r(Class_i) \times IOU_{pred}^{truth} \tag{2.5}$$

Finally, the algorithm uses non-maximum suppression for each class such that only the bounding box with the highest confidence is used.

The network

The network structure in the original paper (Redmon et al., 2016) consists of 24 convolutional layers followed by 2 fully connected layers (see figure 2.5). In the third version however, a network of 53 convolutional layers are used, as depicted in figure 2.6. For dimension reduction it uses 1 x 1 layers which are followed by 3 x 3 convolutional layers.

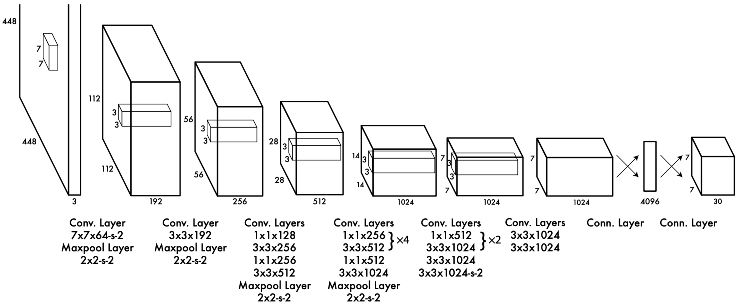
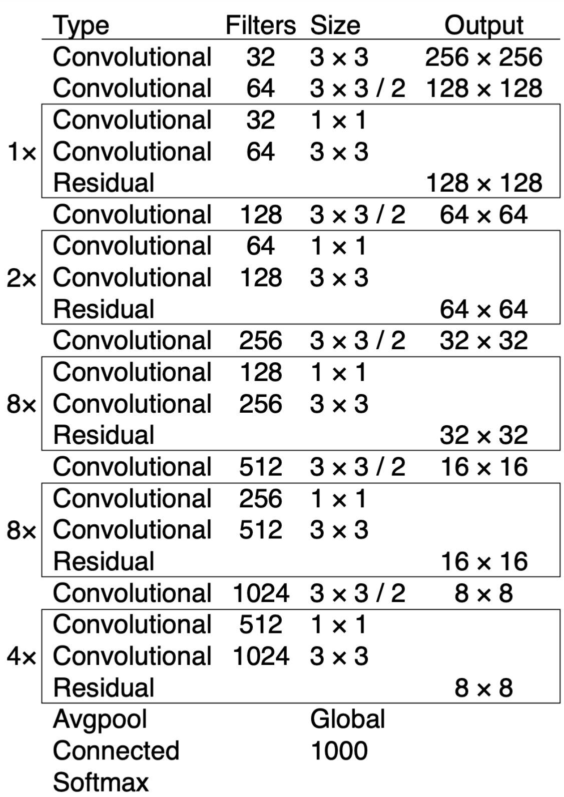


Figure 2.5: Visualisation of the network structure in the original YOLO-paper by Redmon et al. (2016).



	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

Figure 2.6: Visualisation of the network structure in YOLO version 3 Redmon and Farhadi (2018)

2.2.4 Single Shot Multibox Detector

The Single Shot Multibox Detector (SSD) is also a single stage detector and has three main steps (Wei Liu, 2016). The first part of the network extracts features from the image by applying several consecutive convolutional layers in decreasing size. A stack of feature maps with varying sizes are kept from the different convolutional layers, and are used to make detections. The second part of the network produces predictions of bounding boxes and classes in the image. Pre-computed priors (also called anchor boxes), are used for this purpose. Priors are pre-computed bounding boxes with fixed size that closely match the distribution of the original ground truth boxes. In total 1420 priors are used for each image, resulting in robust coverage of objects with different sizes. Finally, non-maximum suppression is applied to only select the most accurate bounding box for each detection.

2.2.5 RetinaNet

RetinaNet's main difference from the other one stage detectors is that it uses focal loss to address the scenario where there is an extreme imbalance between foreground and background classes during training (Lin et al., 2017). The focal loss reduces the loss from easy examples during draining, because training sets usually contain a much larger number of easy examples which cause the loss from the harder ones to almost be neglected, despite that these examples carry more important signals. The common way to address this imbalance is to use a cross entropy function:

$$CE(p_t) = -\alpha_T log(p_t) \tag{2.6}$$

This paper on the other hand uses a focal loss function:

$$FL(p_t) = -(1 - p_t)^\gamma log(p_t) \tag{2.7}$$

α is a weight factor between zero and one, p is the model's estimated probability for the class and γ is a tunable focusing parameter which is larger or equal to zero. The difference between the cross entropy and focal loss functions is visualised in figure 2.7

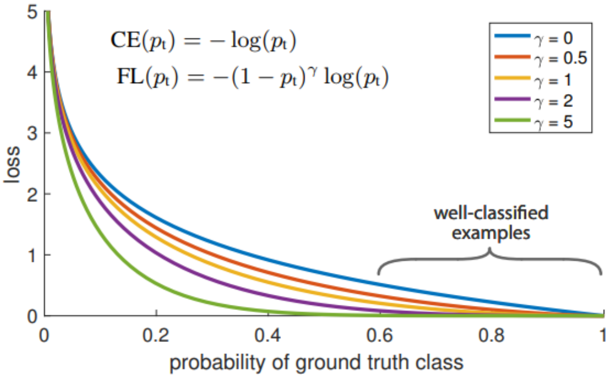


Figure 2.7: Comparison between cross entropy function and focal loss function.

2.3 Range estimation

Monocular vision does not provide range information explicitly. However, the range can be estimated based on the camera's position and pose together with the relevant object's location in the image frame. One such method is presented in Helgesen et al. (2020) and this thesis adopts much of their work. The theory in this paper will therefore be explained in this section.

The algorithm in Helgesen et al. (2020) starts by applying a Single Shot Multibox Detector (SSD) to detect objects in the image. The point of intersection between the detected objects and the ocean surface is then located using a Sobel operator and Hough transform. Then the bearing θ and elevation φ is calculated from the pixel position $\boldsymbol{x}_P^c = \begin{bmatrix} x_P^c & y_P^c \end{bmatrix}$ as follows (see fig 2.8 for visualisation):

$$\theta = \frac{x_P^c - R_x/2}{R_x} F_x \tag{2.8}$$

$$\varphi = \frac{y_P^c - R_y/2}{R_y} F_y \tag{2.9}$$

Where R_i is the image resolution in dimension i, and F_i is the field of view (radians) in the respective dimension. A vector, $\boldsymbol{v}$, pointing at the intersection between the detercted object and the ocean surface, is then created in the camera coordinate system (C) $\boldsymbol{v}^c$:

$$\boldsymbol{v}^c = \begin{bmatrix} \tan(\theta) & \tan(\varphi) & 1 \end{bmatrix} \tag{2.10}$$

This is then transformed to the world coordinate system (w), using the camera position and pose:

$$\boldsymbol{v}^w = \boldsymbol{R}_c^w \boldsymbol{v}^c + \boldsymbol{t}_c^w \tag{2.11}$$

Here, $\boldsymbol{t}_c^t$ is the translation and $\boldsymbol{R}_c^w$ the rotation.

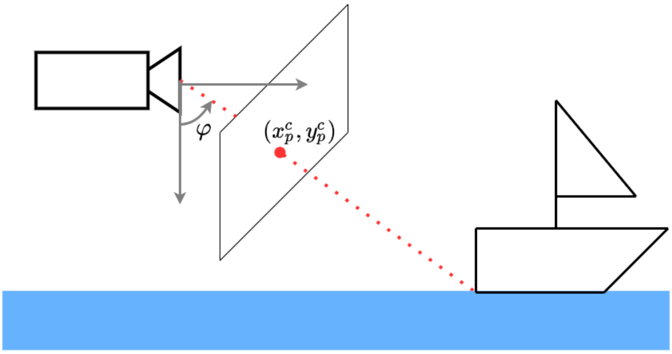


Figure 2.8: Elevation angle is calculated from the pixel position of the detected object.

The vector should end at the intersection between the detected object and the ocean surface (at $z^w = 0$). To achieve this, a scaling factor (s) is calculated from the cameras elevation t^w_{cz}:

$$s = \frac{t^w_{cz}}{z^w} \tag{2.12}$$

The objects position is then calculated with the following equation:

$$\boldsymbol{x}^w = \boldsymbol{t}^w_c + s\boldsymbol{v}^w \tag{2.13}$$

To accurately detect the intersection between the detected object and the ocean surface the Sobel operator and Hugh transform are used. The Sobel operator is a method for detecting edges, and consist of two 3x3 filters convolved with the image. One filter finds the horizontal gradients and the other finds the vertical ones:

$$\boldsymbol{G}_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \tag{2.14}$$

$$\boldsymbol{G}_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \tag{2.15}$$

The Hough transform is a method for detecting geometric features in an image. It is based on the idea that a line in the image has a corresponding point in a parameter space describing a line. The parametrization is given by:

$$\rho = x\cos\theta + y\sin\theta \tag{2.16}$$

x and y is a point i the image, while ρ and θ are points in the corresponding parameter space. A line is then detected by using a two dimensional accumulation array where each cell corresponds to a certain pair of line parameters. If a line is detected close to a pixel, the cell in the accumulation array corresponding to this line is incremented by one.

Chapter 3

Related work

3.1 Object detection in marine environments

Object detection is a vital part of this project, and over the last years it has been an increasingly researched topic. In Grini and Brekke (2019) different object detection algorithms for detecting ships in Trondheimsfjorden were evaluated. The evaluated algorithms included Sliding Window (Sermanet et al., 2013), R-CNN (Girshick et al., 2013), Fast R-CNN (Girshick, 2015), Faster R-CNN (Ren et al., 2017), YOLO (You Only Look Once) (Redmon et al., 2016; Redmon and Farhadi, 2017, 2018) and SSD (Single Shot Detector) (Wei Liu, 2016). Yolo and SSD were further trained on a custom data set and tested in Trondheimsfjorden. It was concluded that YOLO has better performance than the SSD in the experiments conducted.

3.2 Low altitude georeferencing with image sensors in marine environments

This paper is explained in more depth in section 2.3, and its key components are summarised here.

In Helgesen et al. (2020) a method for low altitude georeferencing with image sensors in marine environments were introduced. The Single Shot Multibox Detector (SSD) object detection method was used. More specifically it used a network named Mobilenet v2 (Sandler et al., 2018) which was pretrained on the COCO data set and further trained on a custom data set of 2035 images. A Sobel operator and Hough transform (Duda and Hart, 1972) was used to accurately find the intersection between the object and the ocean surface. A vector from the camera's position to the intersection between the detected object and the ocean surface was found, and the length of the horizontal component of this vector was then the estimated distance to the object.

The system was evaluted on datasets recorded in the fjord outside Oslo, Norway, from a 3 meter tall stationary sensor rig, and benchmarked against a radar set up. Two reference targets were used with on board GPS to record the ground truth. The system was tested with an electro-optical camera, an IR camera and a fusion between the two. It was shown that an electro-optical camera can achieve similar accuracy as a radar at certain ranges during day time.

3.3 Passive target tracking of marine traffic ships using onboard monocular camera for unmanned surface vessel

Park et al. (2015) used a monocular camera and an automatic feature extraction algorithm to detect trafficking ships. The relative bearing was then determined based on the detection. A Canny edge detector and the Hough line transform were employed to detect the horizon in the image. This enabled the algorithm to narrow its search area by only searching below the horizon in the image. Feature from the accelerated segment test (FAST) corner detector (Rosten et al., 2010) was then applied to find target features in the search area. Then Euclidean distance among the features were used in order to detect the desired objects. The position of the detected objects, projected on the image plane considering the camera geometry and the relative coordinates in the observer's reference frame is shown in figure 3.1.

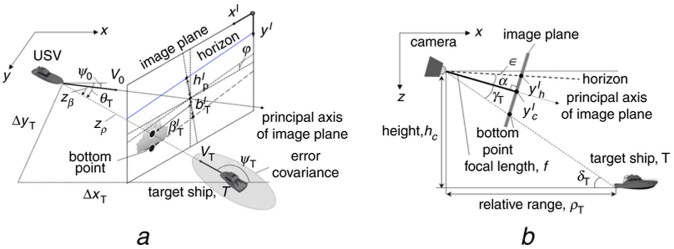


Figure 3.1: Reference frames used in Park et al. (2015)

The relative bearing to the detected object was then calculated using equation 3.1.

$$\beta_T = \frac{F_c}{w_p^I} \beta_T^I \cos \varphi \tag{3.1}$$

Where F_c is the camera's field of view (FOV), w_p^I the width of the image, β_T^I the pixel distance from the target ship to the center line of the image plane and φ is the slope of the horizon.

The distance to the object was then calculated based on the vertical distance between the horizon and the lowest feature point in the object. This was calculated as:

$$\rho_T = h_c / (\tan \delta_T \cos \beta_T) \tag{3.2}$$

where h_c was the height from the horizon to the camera mounted on the vehicle, and ρ_T was determined as:

$$\rho = \begin{cases} \gamma_T + \alpha + \epsilon \approx \gamma_T + \alpha & \text{if } y_h^I \leq y_c^I \\ \gamma_T - \alpha + \epsilon \approx \gamma_T - \alpha & \text{otherwise} \end{cases} \tag{3.3}$$

y_h^I and y_c^I represent the vertical coordinate of the image centre and the horizon. γ_T and α were defined as:

$$\gamma_T = \arctan(\frac{b_t^I}{f}) \tag{3.4}$$

$$\alpha = \arctan(\frac{h_p^I}{f}) \tag{3.5}$$

where f was the focal length. b_t^I and h_p^I was the distance between the centre of the image and the lowest feature in the detected object and the horizon respectively. The target's trajectory is then estimated based on the observations.

The algorithm was tested in field experiments using an electro-optical camera mounted on a USV. The results show an increased state observability compared to when only bearing was used for estimation (see figure 3.2).

3.4 Stereo vision

Stereo vision utilizes two cameras positioned parallel to each other, and extract depth information based on the distance between the two cameras and the detection point of an object in each image from the two cameras. A visualization of the concept can be seen in figure 3.3. Equation 3.6 and 3.7 can be used to calculate the projection of a point P to the image planes, and then the disparity can be calculated with equation 3.8. Depth information can then be extracted from the disparity (Gul et al., 2021).

$$u_L = f\frac{X_A}{Z_A} \tag{3.6}$$

$$u_R = f\frac{X_A - b}{Z_A} \tag{3.7}$$

$$d = (u_L - u_R) = f\frac{b}{Z_A} \tag{3.8}$$

Where u_L and u_R are the position of a real world point P in an image captured by the left and right camera respectively. f is the focal length of the cameras, b is the baseline, X_A is the X-axis of the camera and Z_A is the optical axis of the camera.

In Auestad et al. (2021) the effectiveness of stereo vision methods on marine vessels in the Trondheims fjord was evaluated. Different techniques for solving the correspondence problem was also examined. The correspondence problem is the task of recognising the same point in both images. The methods for solving this problem were divided into local and global methods. Among the local methods the following were evaluated: Sum of

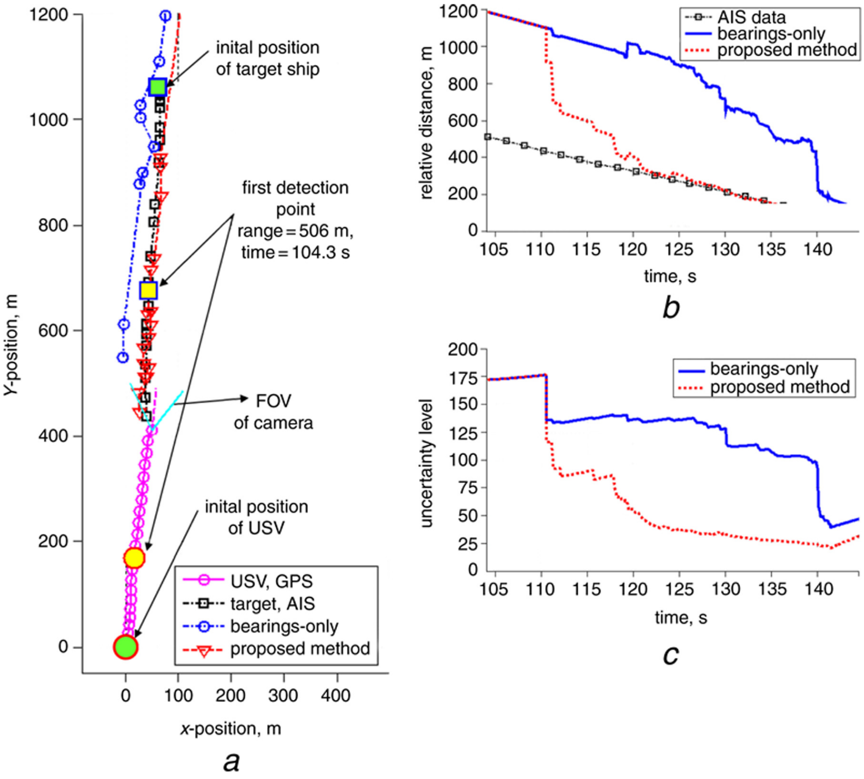


Figure 3.2: Results from tracking algorithm in Park et al. (2015)

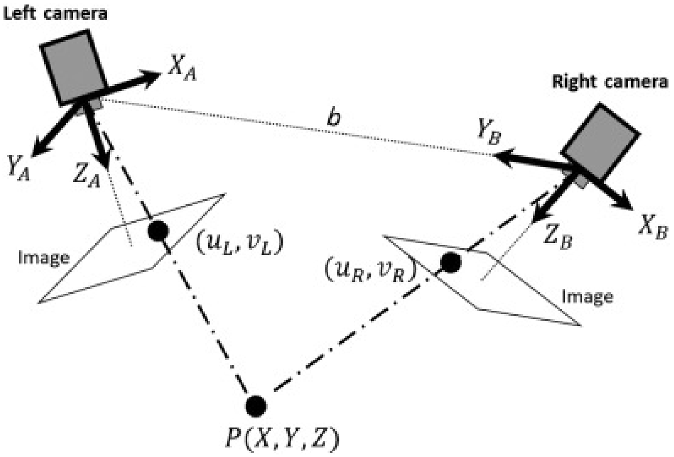


Figure 3.3: Visualisation of the consepual working stereo vision. Source: Gul et al. (2021)

Absolute Differences, Sum of Squared Differences, Normalized Cross Correlation, Rank Transform, Census Transform, Scale Invariant Feature Transform, Speeded Up Robust Features and Oriented FAST and Rotated BRIEF. The following global methods were also tested: Dynamic Programming, Graph Cuts, Semi-Global Method, Pixelwise Cost Calculation and Aggregation of Cost. It was concluded that the Sum of All Differences (SAD) and Semi-Global Method (SGM) were the most promising methods, but that SGM had the best trade off between accuracy and run time among the evaluated methods.

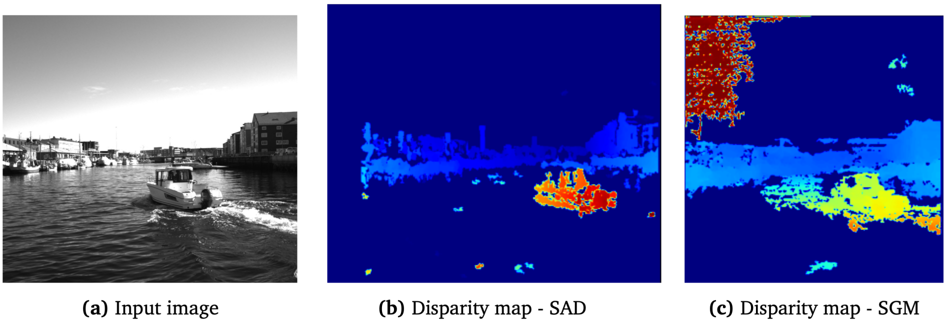

(a) Input image **(b)** Disparity map - SAD **(c)** Disparity map - SGM

Figure 3.4: Disparity map comparison between SAD and SGM. Source: Auestad et al. (2021)

Five object detection algorithms were also tested: Stixel Tesselation Badino et al. (2009), Digital Elevation Map Sabbatelli et al. (2014), Geometry-based Cluster Talukder et al. (2002), Direct Planar Hypothesis Testing (DPHT) Pinggera et al. (2015) and Euclidean Clustering. It was concluded that the DPHT method shows great promise in literature, but was dropped due to its lack of open source code. The Sitxel-method on the other hand is well documented and has open source experiments available. It also showed good performance regarding accuracy and run time in the literature and was therefore chosen by the author.

The stereo vision system was tested in the Trondheims fjord, and the results compared to the results of a lidar. It was concluded that the stereo vision system had lower accuracy but was able to operate on longer distances than the lidar.

Chapter 4

Methods

4.1 Hardware

USV

The USV used in the project was based on the Otter (MaritimeRobotics, 2022) developed by Maritime robotics. The vehicle is $200cm$ in length, $108cm$ wide, $106.5cm$ tall. It was powered by two electrical fixed thrusters, giving it a top speed of 6 knots. The on board computer was a Raspberry Pi Compute Module 4 with 8GB RAM, 32GB eMMC and WIFI/BT. For storage a 512GB NVMe PCIe Solid State Drive was used

The USV was also equipped with a GPS and a camera.

Camera

As part of this thesis, different camera options were researched and compared. The requirements for the camera was mainly the following:

- Low cost (maximum 15 000 NOK).
- Water and shock resistant
- Night/low light capability
- Connectable with PoE (Power over Ethernet)

Thermal cameras within the price range were extensively searched for, however none was found. Stereo cameras were also considered, but discarded due to a desire to keep the solution simple and concerns regarding the run time of available algorithms. It was therefore decided to choose a camera with IR lighting. Several options were considered and a Hikvision DS-2CD2343G2-I(U) was ultimately found to be the best fit. This camera has the following relevant specifications:

- IR light range: 30 meters

- Minimum illumination: 0.005 Lux
- Water resistance: IP67
- Temperature range: -30 °C to 60 °C
- Connectable with PoE (Power over Ethernet)
- Horizontal FOV: 103°
- Vertical FOV: 55°

Figure 4.1: Hikvision DS-2CD2343G2-I(U). The selected camera for this thesis.

4.2 Algorithm

An algorithm heavily inspired by Helgesen et al. (2020) was implemented. The implemented algorithm however differed in a few ways which will be described in this section.

The algorithm works in three steps:

1. An object detection algorithm detects objects and returns bounding boxes describing the objects location in the image.
2. From the bounding boxes, the bearing and elevation angle to the detected objects are calculated.
3. Distance to the detected objects are calculated based on the bearing, elevation angle and the cameras position.

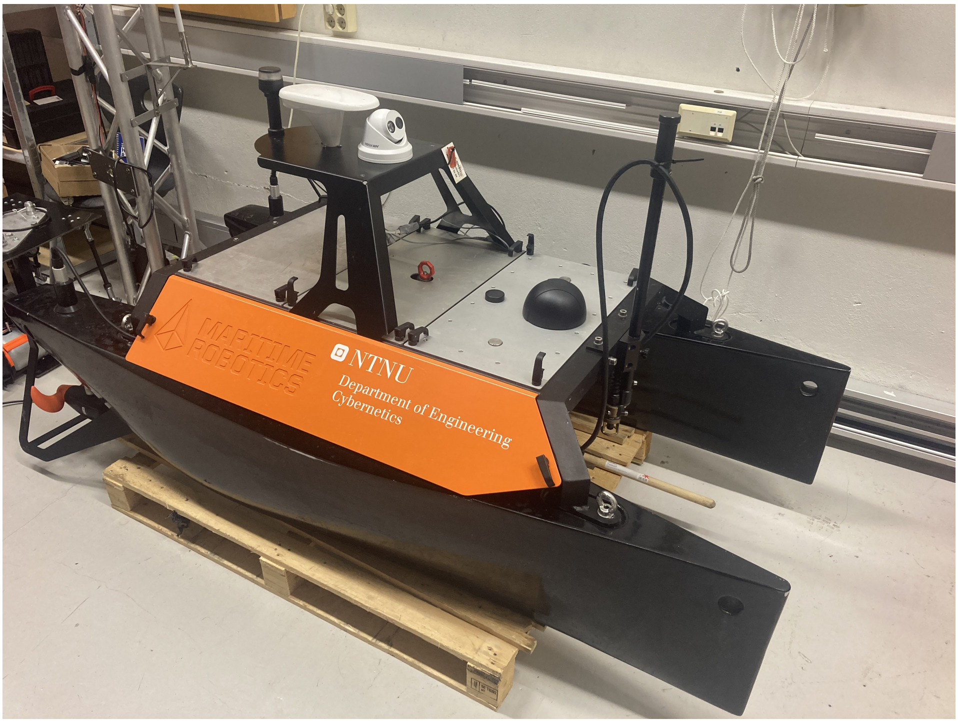


Figure 4.2: The USV used in the thesis, with the chosen camera mounted.

The algorithm chosen for object detection was the YOLO darknet version 3 (Redmon and Farhadi, 2018), and the decision was made based on the information found during the literature review. It was desired to have a algorithm with as low run time as possible such that it could be run at the USV's computer, which has fairly limited computational power. This excluded the two stage detectors like R-CNN and Slidingwindow, and among the single stage detectors evaluated, YOLO had the lowest run time. Furthermore, there was put much emphasis on the conclusion in Grini and Brekke (2019), since this was based on an experiment in very similar condition to the one conducted in this project. Grini and Brekke (2019) concludes that YOLO has higher performance than SSD in these conditions. The YOLO algorithm was also available open source and was easy to download and run. In the YOLO paper (Redmon and Farhadi, 2018), a threshold of 0.5 for the IOU was used, but from the experiment conducted in this thesis it was found that a threshold of 0.1 was favourable for this application. This caused the algorithm to frequently errantly classify boats as cars and docks as boats, but with a threshold of 0.5 the algorithm did not detect these objects at all. Given that the method was intended to avoid collisions, the main objective of the object detection algorithm was to detect objects in the USVs path and whether the object was a boat or a dock was of less importance. From the experiment, the lowered threshold did not cause any waves or similar to be wrongly classified as objects, which would have been an issue.

From the bounding boxes returned by the YOLO-algorithm the lowest point in the vertical direction and the midpoint in the horizontal direction of each bonding box was extracted. Based on this point, the bearing (θ) and elevation angle (φ) was calculated in the same way as in Helgesen et al. (2020):

$$\theta = \frac{x_P^c - R_x/2}{R_x} F_x \tag{4.1}$$

$$\varphi = \frac{\pi}{2} - (\frac{y_P^c - R_y/2}{R_y} F_y - \phi_c) \tag{4.2}$$

Where ϕ_c is the camera angle equal to 15°. The camera was mounted with an angle of 15° because this was the lowest angle possible, due to the structural design of the camera. Distance was only estimated to objects detected with an $\varphi \leq 90$. This is because, with the assumption that all detected objects are in the water-plane, objects with $\varphi \geq 90$ are infinitely far away. However the method described below will estimate them at the same distance as if φ was equal to $180 - \varphi$, which gives misleading results.

From θ and φ a vector in the camera coordinate system, pointing at the detected object was created:

$$\boldsymbol{v}^c = \begin{bmatrix} \tan(\theta) & \tan(\varphi) & 1 \end{bmatrix} \tag{4.3}$$

This vector was then scaled by a scaling factor equal to the height (h) of the camera. The distance to the object was then described as:

$$distance = h \times \sqrt{\tan\theta^2 + \tanh\varphi^2} \tag{4.4}$$

Unlike in Helgesen et al. (2020) a Hough transform and Sobel operator was not used to detect the intersection between the detected objects and the ocean surface. The reason for this decision was a desire for simplicity in the algorithm, and will be discussed in chapter 6.

4.3 Experiment

An experiment was conducted in the Trondheims fjord in the marina at Børsa close to Trondheim. The goal of the experiment was to record a data set containing video and GPS-coordinates from the USV, and test the algorithm on this data set. The USV was placed on the water from the marina and controlled using a Bluetooth controller connected to a computer which communicated with USV through wireless 4G network. Video from the camera was stored on an local micro SD card with 128 GB storage capacity. Data from the GPS was stored on the USVs computer. In order to find the ground truth distance to observed objects, the USV was driven to the relevant objects, namely boats, docks and buoys. By doing this, the GPS-coordinates of their locations were obtained and later used to measure the distance between these objects and the USV. To find the position of the USV for a given frame, the timestamp of the camera was matched with the timestamp of each data point from the GPS. The GPS data was loaded to a open source program called QGIS. Here each GPS point was plotted against an open source map, and the ruler tool was used to measure the distances between the USV and the detected objects. The estimated distances from the algorithm was then compared to the ground truth from the GPS.

When evaluating the success of the algorithm, a set of images was extracted from the recorded data set and fed to the algorithm running on an external computer. The distance to the relevant objects were logged in a table together with the corresponding distances found using the GPS. Some of the detected objects were very similar and located close to each other. These objects were therefore grouped and regarded as one in the evaluation process. The median of the estimated distances to the grouped objects was then used during evaluation.

The results were evaluated using the Root Mean Square Error (RMSE) method which is defined as:

$$\boldsymbol{RMSE} = \sqrt{\frac{\sum_{i=1}^{k}(\hat{x}_i - x_i)^2}{k}} \tag{4.5}$$

Where $\hat{x}_i$ is the estimates, x_i is the true values and k is the number of measurements. Objects detected on land were not included in the evaluation process. This is because they are not located in the water plane, and therefore does not fulfil this key assumption in the estimation method. Objects on land will have a higher elevation in the image, and are therefore estimated to be further away than they are. The USV however, is able to avoid land using GPS and a map, and this was therefore not regarded as an important issue.

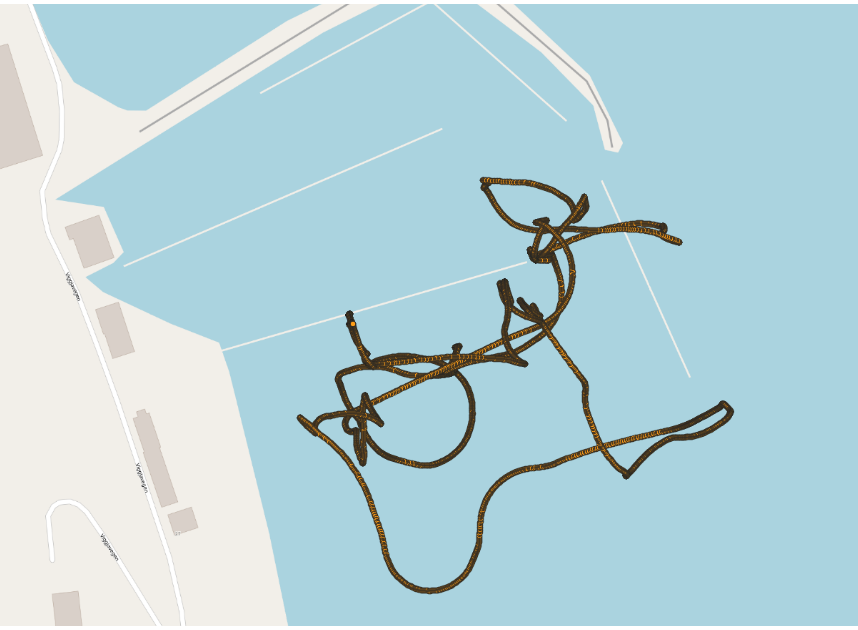


Figure 4.3: GPS points from the experiment plotted against an open source map in QGIS.

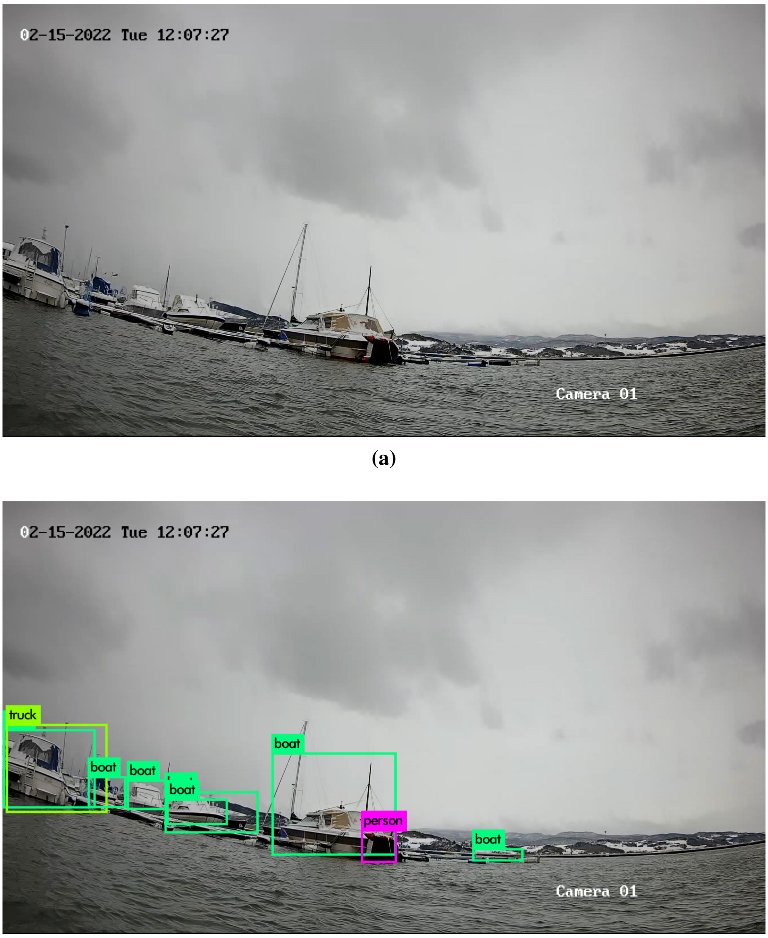
Chapter 5

Results

In this chapter the results from the experiments are presented.

What	Ground truth distance (m)	Estimated distance (m)	Error (m)
$Picture1$			
Rightmost dock	23	8.7	-14.3
Boat middle	17	11.3	-5.7
Boat left	15	96.2	81.2
$Picture2$			
Rightmost boat	24	11.8	-12.2
Dock middle	19	9.8	-9.2
Boat left	23	16.7	-6.3
$Picture3$			
Sailboat	33	61.5	28.5
Rightmost dock	24	43.7	19.7
$Picture4$			
Buoy	5	5	0
Boats in the centre of image	31	22,5	-8,5
Boats to the right	60	23	-27
$Picture5$			
Boats in background	80	135.3	55.3
$Picture6$			
Boats in the centre of image	16	32	16
Boats to the right	60	614	554
$Picture7$			
Buoy	3.2	11.5	8.3
Land	34	14.5	-19.5
$Picture8$			
Large boat	24	25.8	1.8
car	69	23	-46
Boats to the right	52	21,1	-30.9
$Picture9$			
Sailboat	24	9.8	-14.2
Boat in the middle	26	51.5	25.5
$RMSE$			
All			124.2
Ground truth distance $\leq 20m$			34.2

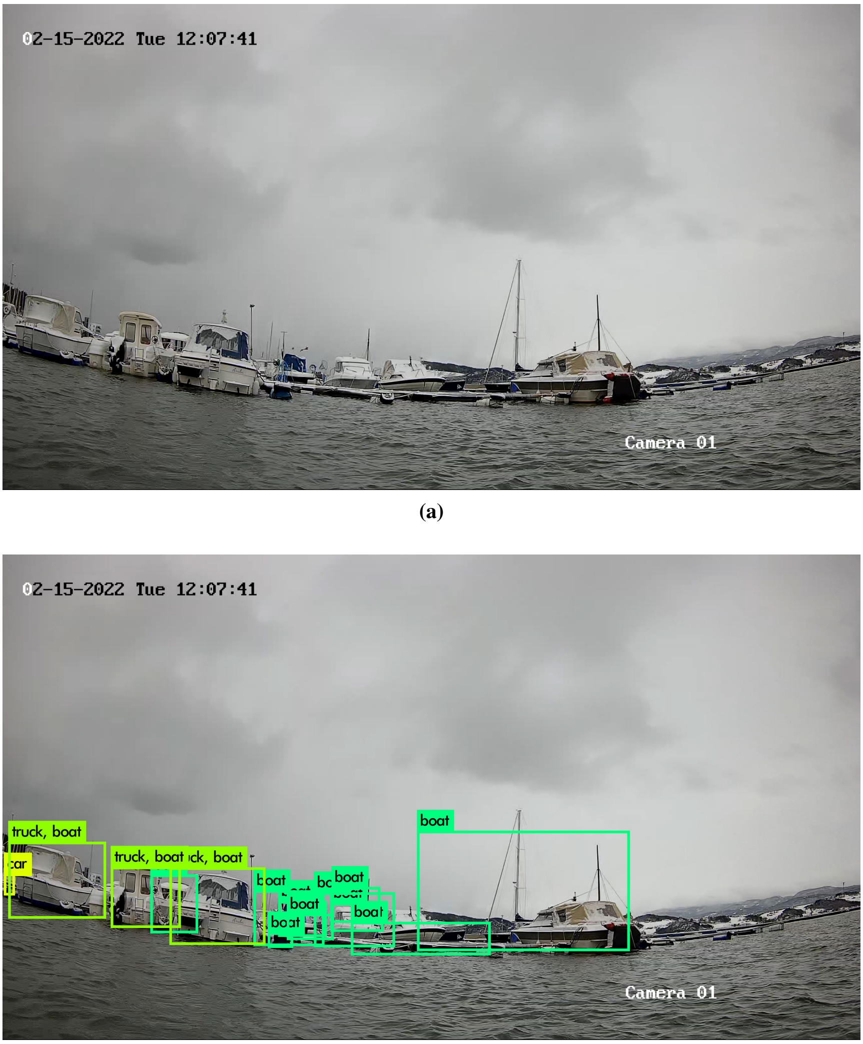
Table 5.1: Table summarising the results. Each section corresponds to one image extracted from the recorded data set.



(a)

(b)

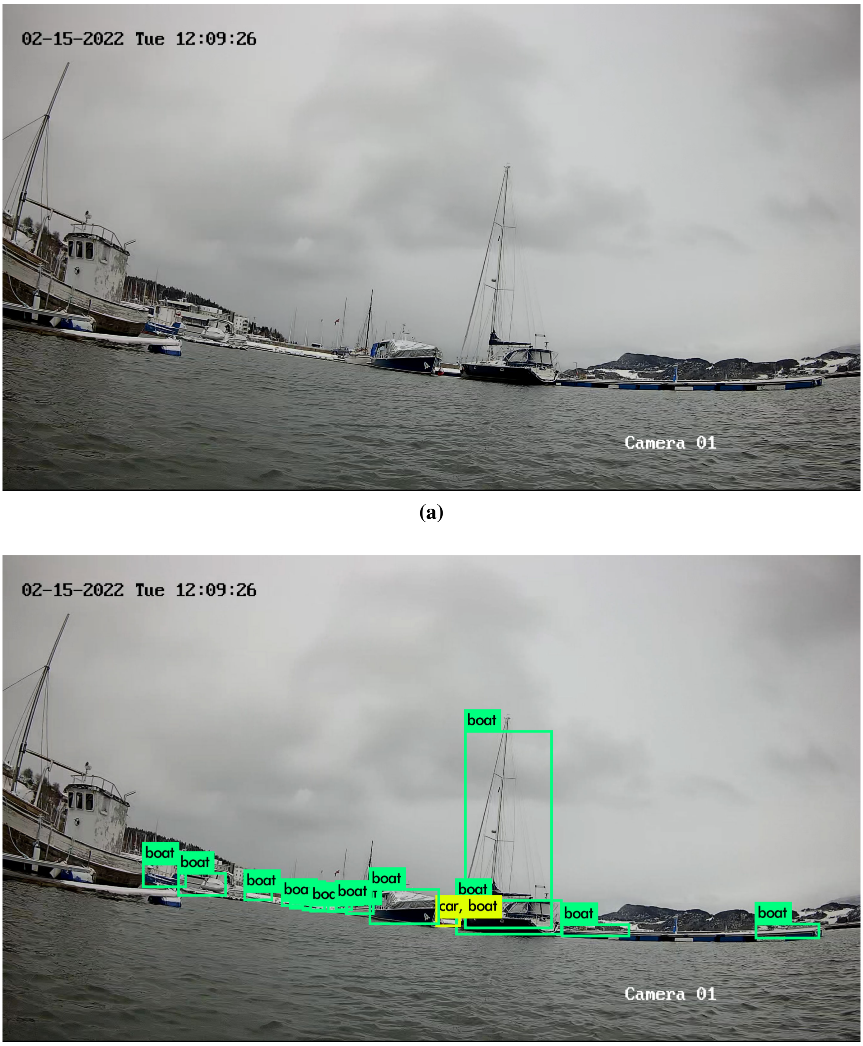
Figure 5.1: Picture 1
First picture extracted for evaluation.



(a)

(b)

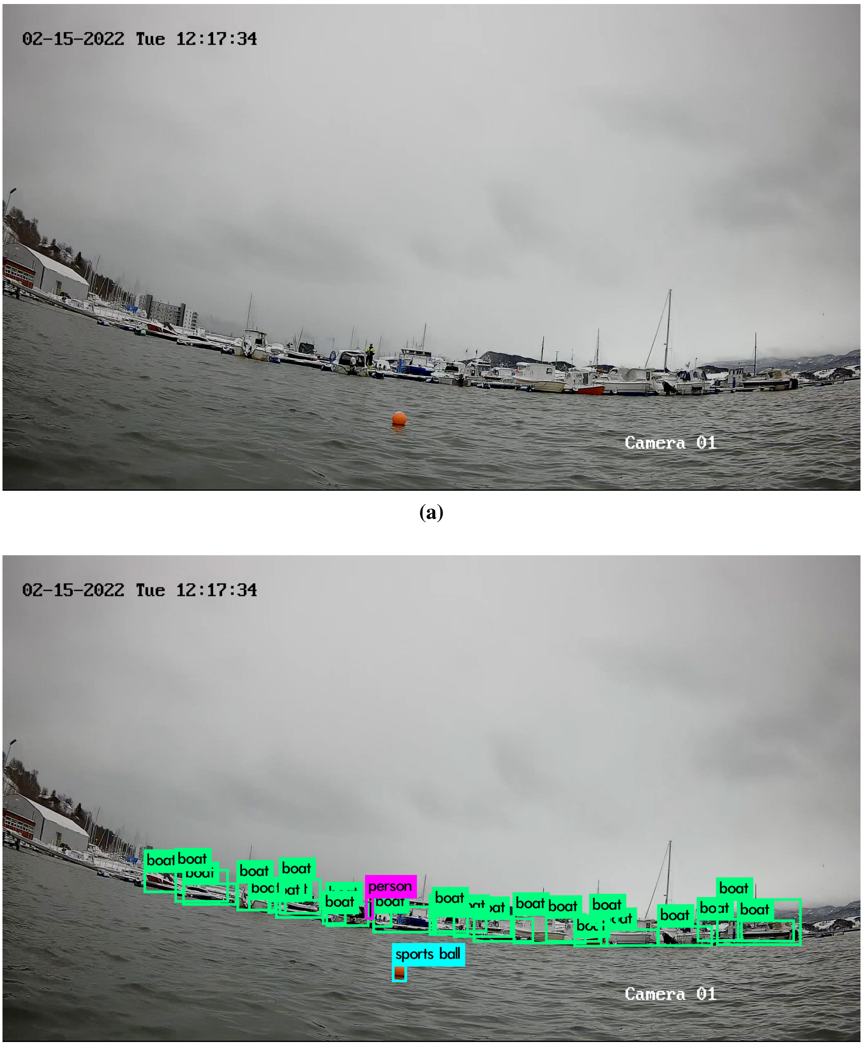
Figure 5.2: Picture 2
Second picture extracted for evaluation.



(a)

(b)

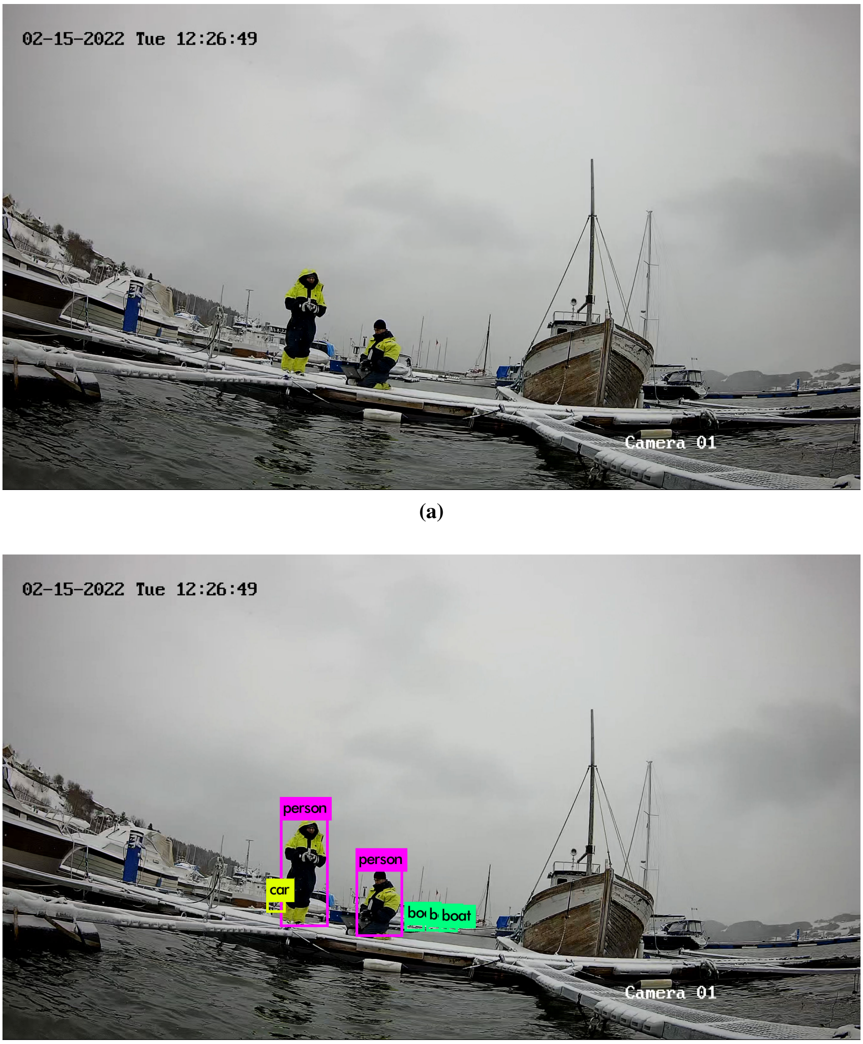
Figure 5.3: Picture 3
Third picture extracted for evaluation.



(a)

(b)

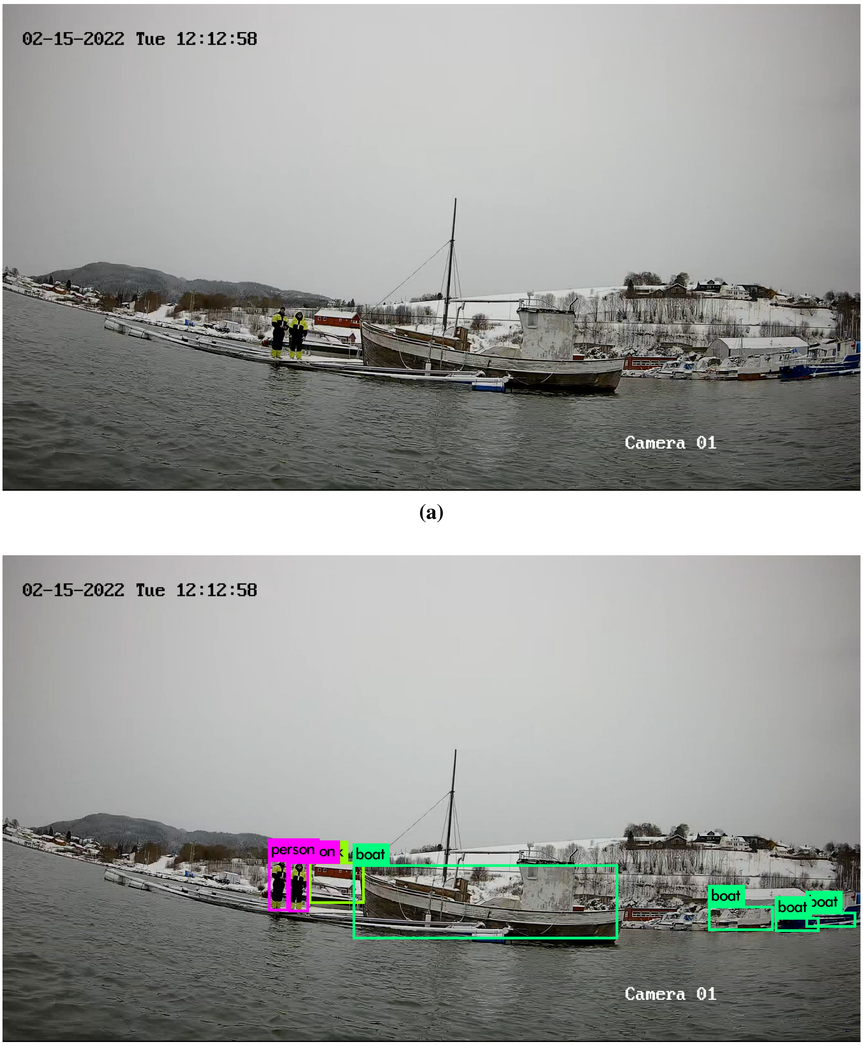
Figure 5.4: Picture 4
Forth picture extracted for evaluation.



(a)

(b)

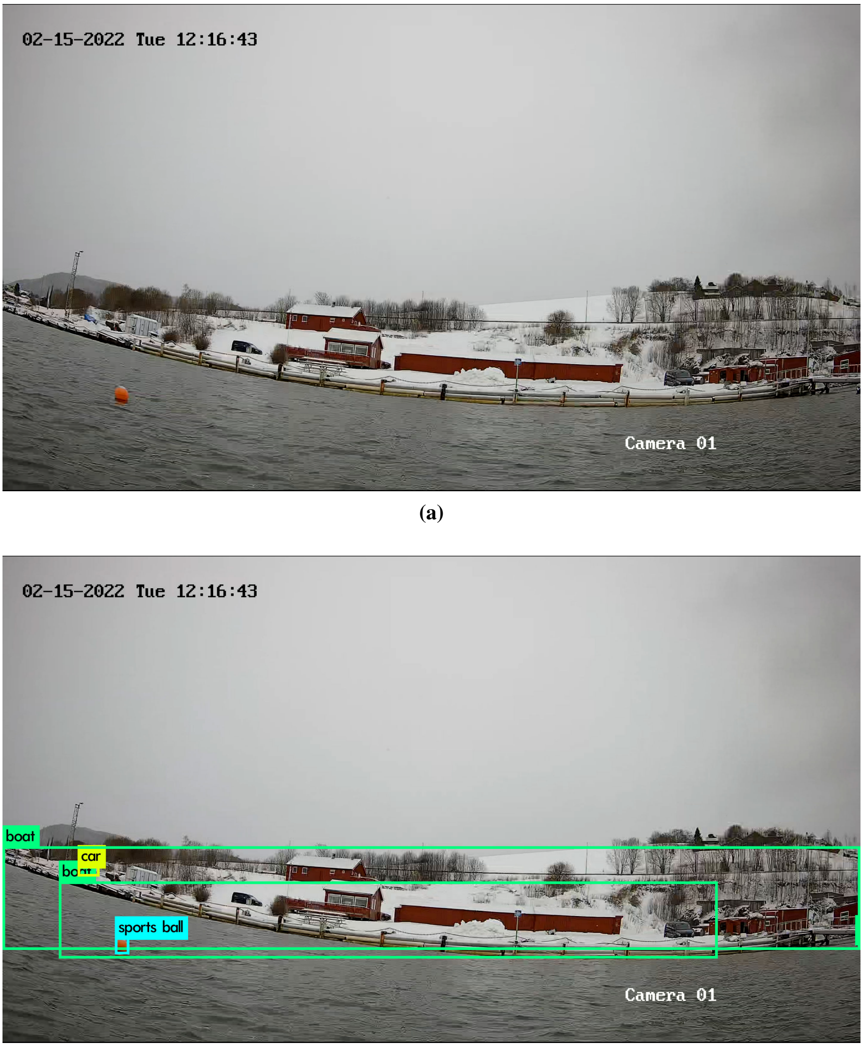
Figure 5.5: Picture 5
Fifth picture extracted for evaluation.



(a)

(b)

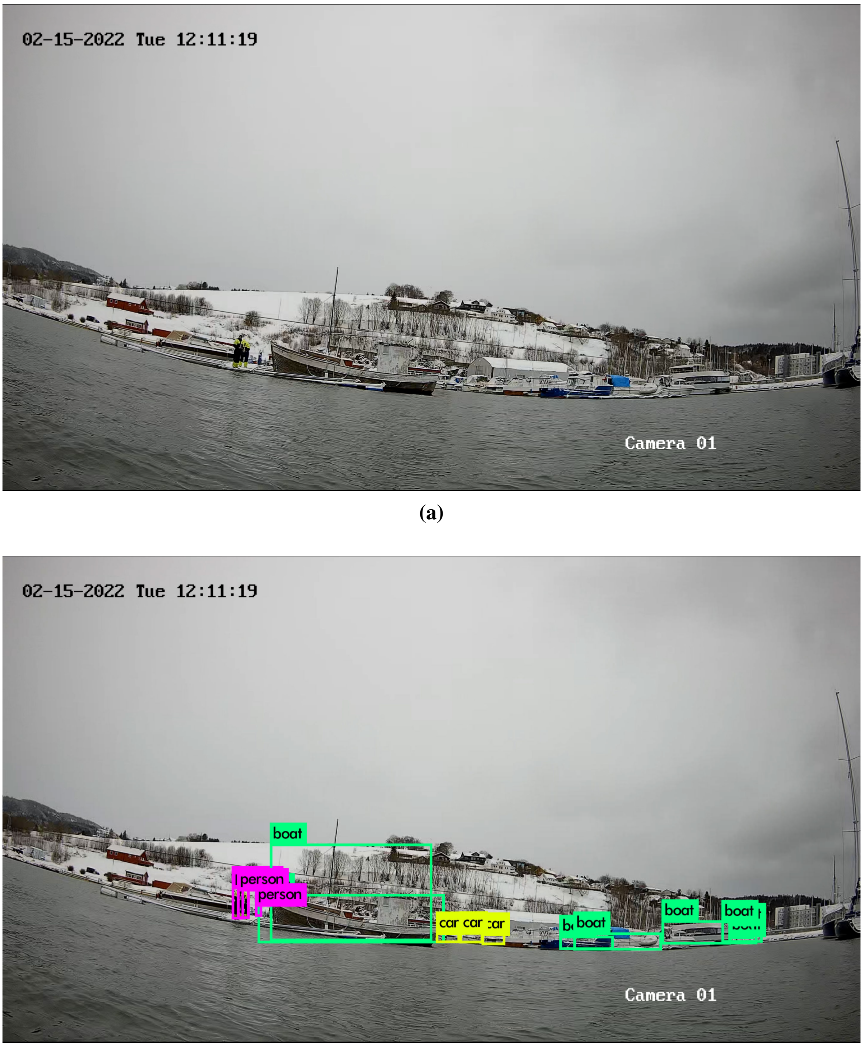
Figure 5.6: Picture 6
Sixth picture extracted for evaluation.



(a)

(b)

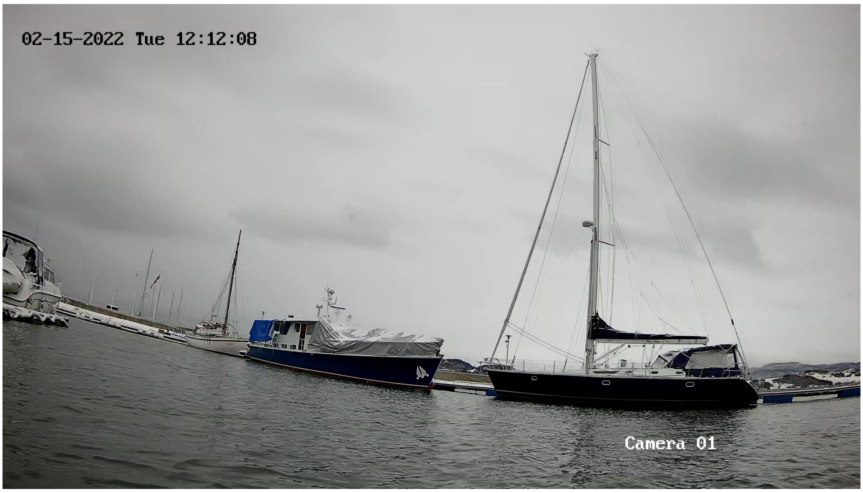
Figure 5.7: Picture 7
Seventh picture extracted for evaluation.



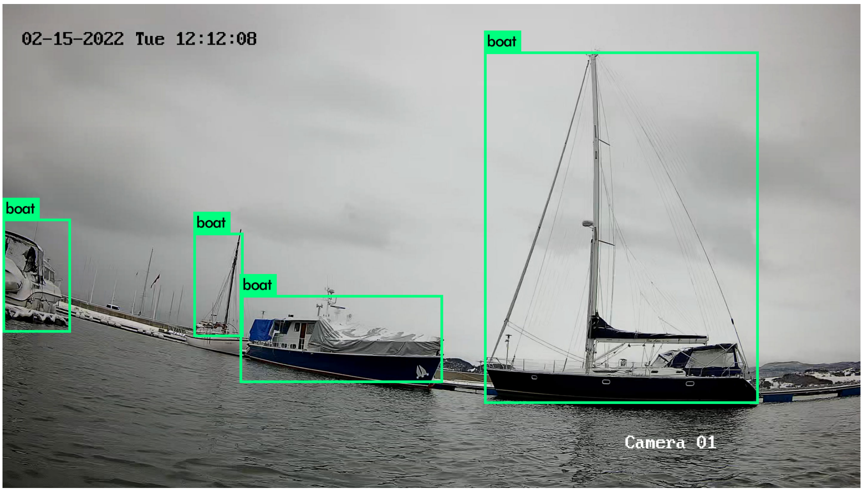
(a)

(b)

Figure 5.8: Picture 8
Eighth picture extracted for evaluation.



(a)



(b)

Figure 5.9: Picture 9
Ninth picture extracted for evaluation.

Chapter 6

Discussion

In this chapter the results will be evaluated. In addition, the viability of the system and possible improvements will be discussed.

Accuracy

From table 5.1 we see that the RMSE is quite high relative to the actual distance to the objects, which is not ideal. When looking closer at the data we observe that a few measurements contribute disproportionately much to this error. Most notably the estimates to the rightmost boats in Picture 6 (figure 5.6) greatly contribute to the overall RMSE. Similarly, the estimates to the boat located to the left in picture 1 (figure 5.1) contribute greatly to the RMSE for objects closer than 20 meters. This can therefore partly explain the high error. However, even without picture 6 and 1, the error in the remaining images gives reason for some concern, and is probably too high to serve as the basis of a collision avoidance system. Despite this, the algorithm does in most cases give a reasonable ”ballpark” estimation of the distance to the objects, and with some improvements it could prove useful.

Wave sensitivity

The system is highly sensitive to roll and pitch induced by waves. To illustrate the magnitude of this problem, consider the following scenario: If an object is detected straight ahead at an elevation angle at 88.5° the estimated distance would be 19 meters. A small wave causing the USV to pitch 1° downwards would change this estimate to 57.3 meters. Similar problems would occur in roll, causing objects on one side of the image to be estimated to be way too close and the opposite side to be way too distant. This problem however becomes less severe the closer the object is to the USV. To illustrate the difference, consider an object detected straight ahead with an elevation angle at 70°. The system would estimate it to be 1.37 meters away, and a 1° pitch downwards would only change this to 1.45 meters (a 8 cm difference). Even a 10° downwards pitch would only change the estimated distance to 2.84 meters. Still this is a major disadvantage with the system, and efforts should be made to mitigate this issue. One possible way of dealing with this could be to estimate the pitch and roll, and compensate for this when calculating the distance. Pitch and roll can be estimated using an IMU (Internal Measurement Unit), or by tracking each object over some time period and utilizing the oscillating angles retrieved from the bounding boxes to make the estimates. The problem can also be solved by detecting the horizon and using this as a reference. This will be further discussed later in this chapter.

Image distortion

When inspecting the data set, a distinct distortion in the image can be observed. This is a common phenomena in optical sensors. The distortion causes objects at the edges to appear higher in the image than they should, causing the algorithm to estimate a too large distance to them. It also causes the appearance of objects straight ahead to be distorted. This is a critical and unnecessary source of error, and should be corrected in future work.

Lack of Hough Transform and Sobel Operator

As mentioned earlier, this work is heavily influenced by Helgesen et al. (2020), and implements most of the methods used in that paper. This work does however not include an Hough Transform nor a Sobel Operator to increase the accuracy of the located intersection between the detected objects and the ocean surface. This was a choice made based on a desire to keep the system as simple as possible. However in (Helgesen et al., 2020) it is emphasised that finding the accurate point of intersection is key to obtaining high accuracy estimates using this method. That paper however tested the method on objects up to 400 meters away. The camera is also mounted stationary on land which liberates it from the interference of waves. It is therefore tested on a different basis than in this thesis, and also achieves higher accuracy than what is needed in this project.

Still, when inspecting the results it is obvious that the object detection algorithm does not always provide bounding boxes which perfectly end at the intersection between the objects and the ocean. It is however fairly accurate in most cases, and although this could lead to large deviations for objects more than 20 meters away from the USV, it is not likely to be a source of critical error for objects close to the USV, which is most crucial for this project. A more likely source of critical error would be the interference of waves and the failure to detect objects. In the authors view it would therefore be desirable to investigate the accuracy gains by applying a Sobel Operator and Hough Transform, but there are probably other measures which would increase performance more efficiently.

Object detection accuracy

When inspecting the results we observe that the object detection algorithm in many instances errantly classified objects. Most commonly it wrongly classified boats as cars, trucks or airplanes and docks as boats. It also classifies buoys as sport balls, but given their similarity in both appearance and physical structure this is neglected. The errant classifications are however not viewed as a crucial issue, because the main objective of this system is to facilitate collision avoidance. It is therefore not crucial information if if is a boat, dock, air plane, truck or car the algorithm detects. Given the situations the USV will operate in, it can in most cases be assumed that airplanes, cars and trucks in reality are boats or similar objects. In future iterations however, the project might develop to encompass a higher level of autonomy, and in this case the class of detected objects might become more essential information. A more urgent concern is that the object detection algorithm fails to detect several objects, most notably some boats and docks. This could be a reason for concern if it is to be the only sensor used for collision avoidance. The algorithm does detect the vast majority of objects of interest, but this is still an issue which should be addressed in future work. One possible solution could be to further lower the threshold for IOU in the algorithm. Another could be to test other object detection algorithms to examine if there are better options for this part of the system. A third option is to further train the algorithm on more similar situations to the ones encountered in this thesis. A possible issue with lowering the threshold of the algorithm could be that ocean waves can be wrongly classified as objects, especially during high wind conditions. Some object detection algorithms are purely based on features (like edges and corners in the image) and the euclidean distance between them. These algorithms are often more sensitive to this issue. A way to possibly mitigate this issue would be to to limit the YOLO algorithm to certain classes, such that it only classifies e.g boats, docks, people and buoys.

Horizon detector

A different angle of attack at the problem of range estimation with monocular cameras is to use an edge detector to detect the horizon. The angle between the horizon and the intersection between detected objects and the ocean surface is then used to make a range estimation. As mentioned in Chapter 2, this method is explored in Park et al. (2015). In that paper however the system was tested on a much larger vessel, allowing for a higher position of the camera and less interference from waves. The system is also dependent on detecting the horizon, making it more vulnerable to conditions with low visibility. In areas close to shore, it can also risk wrongly classifying the shore as the horizon, leading to inaccuracies. The method is however less sensitive to the impacts of waves, since it does not depend on the angle between the camera's pose and the object. It would therefore be interesting to compare the accuracy of the two methods in this thesis. Finding a way to combine both methods could also potentially lead to more accurate results.

Stereo Vision

Stereo vision was one of the methods considered for this thesis, it was however not chosen due to a desire for simplicity in the system, and concerns regarding the run time of stereo vision algorithms. Stereo vision though, is a proven method for georeferencing using cameras. It does however require two cameras to function. Fortunately, the cost of the camera chosen for the project is low, and the financial cost is therefore not an issue. It does however take up more space on the USV, which is a scarce commodity. At the time of writing though, there is still enough space for another camera at the same platform as the current camera is mounted. The accuracy of a stereo vision system is highly dependent on the distance between the two cameras (the baseline). Since the USV is small, the baseline would be relatively short. This however would likely not be a large issue, since the system does not require high accuracy at long distances. It would also be possible to build additional structures to accommodate a larger base line if necessary, but this would be a less favourable solution. Lastly, a second camera would require another connection to the on board computer and electronics. This would most likely require the making of a new hole in the box encompassing the electronics. This hole would also need to be waterproof to protect the electronics inside. Although requiring a fair bit of work, this could be done. Because of the issues mentioned above though, a stereo vision system would not be the natural next step in this project. However if other more easily implemented methods do not provide the required accuracy, a stereo vision system could be an ideal addition to the system. This is because it is low cost and could complement existing methods based on monocular cameras. Having two cameras mounted on the USV could in theory also be used to estimate the roll of the vessel, further increasing the accuracy of the method used in this thesis.

System evaluation

As the system is implemented now, it is not ready for deployment. More work is needed both to increase the accuracy, either through upgrades to the currently used method, or by replacing or complementing it with another method. The algorithm is also not implemented in a way that allows it to be run in real time. At its current state it is only capable of running on a pre-recorded data set, and hence only works for demonstration purposes. It does however show some promise in being a first step towards a collision avoidance system. Although the accuracy needs to be improved and some issues need to be solved, it is able to detect most objects of interest. It also provides a reasonable ballpark range estimate and the bearing angle to the objects.

Chapter 7

Conclusion

The system presented in this report does not provide sufficient accuracy to serve as a basis for collision avoidance. It does however show some promise, since it is able to detect most objects of interest and provide a ”ballpark” range estimate to those objects. With some improvements it is possible that it could serve as a basis for a colission avoidance system. Several suggestions for future work to improve the system are listed below.

7.1 Future work and continuation

During this work several ways to improve the system have been discussed and proposed. This section summarises these suggestions in a structured manner.

Firstly, there are several ways to improve the accuracy of the current system, and the following approaches should be pursued:

- The first and most pressing improvement is to rectify the images such that they are not distorted. This will lead to a more accurate result especially for the objects located at the edge of the images.
- The second suggestion is to estimate the roll and pitch of the USV and correct for this in the algorithm. This can be done by using an IMU, a horizon detector, or by tracking the objects over some time and base the estimate on the oscillating angle. A combination of the three might also be an option.
- A third way to increase the accuracy of the current system would be to use a Sobel Operator and a Hough Transform to more accurately detect the intersection between the detected objects and the ocean. This method is described in more depth in Helgesen et al. (2020)
- Training the existing object detection algorithm on a custom data set captured in a similar environment could also improve the performance of the algorithm.

- Lastly, other object detection algorithms should be tested with this setup. Although the literature evaluated in this thesis suggest that YOLO is the best option, it is possible that this is not the case when testing it in practice.

Other methods for estimating distance to objects should also be tested. Most notably, an implementation of the passive target tracking algorithm described in Park et al. (2015) should be examined. Stereo Vision is also a method which could be implemented either instead of the existing method or in combination with it. Stereo vision has the advantage of being a thoroughly tested technique, but falls behind in that it is computationally heavy and requires a second camera to be mounted.

The system should also be further developed such that it can be run real time while the USV is operating on the water. It is also desirable to give some kind of signal to an operator, or stop the USV if it detects an object on collision course with the USV. Using the Robotic Operating System (ROS) would also be an advantage when further developing this system. This is because it is designed to manage sensor data and control signals in real time.

A final suggestion for future work is to test if it is beneficial to run the system locally on the on board computer. The alternative would be to send sensor data to an external computer, using the 4G network, and do all the processing remote. The benefit of doing the processing remotely is that you can use a more powerful computer and hence use a more computationally heavy method. On the flip side, if the 4G connection is lost you have no system running at all.

Bibliography

Albawi, S., Mohammed, T.A., Al-Zawi, S., 2017. Understanding of a convolutional neural network, in: 2017 International Conference on Engineering and Technology (ICET), pp. 1–6. doi:10.1109/ICEngTechnol.2017.8308186.

Alfredsen, J.A., . 1.nbsp;nbsp; robotic fish tracking - integration of auv/usv and acoustic fish telemetry. URL: https://folk.ntnu.no/alfredse/Forslag%20til%20prosjektoppgaver%20hoesten%202019.htm.

Auestad, K., Brekke, E.F., Stahl, A., Helgesen, K., 2021. Depth estimation and object detection using stereo vision for autonomous ferry. NNU Norwegian University of Science and Technology, Faculty of Information Technology and Electrical Engineering, Department of Engineering Cybernetics .

Badino, H., Franke, U., Pfeiffer, D., 2009. The stixel world - a compact medium level representation of the 3d-world. Springer Berlin Heidelberg .

Bre, F., Gimenez, J., Fachinotti, V., 2017. Prediction of wind pressure coefficients on building surfaces using artificial neural networks. Energy and Buildings 158. doi:10.1016/j.enbuild.2017.11.045.

Campbell, S., O'Mahony, N., Krpalcova, L., Riordan, D., Walsh, J., Murphy, A., Ryan, C., 2018. Sensor technology in autonomous vehicles : A review, in: 2018 29th Irish Signals and Systems Conference (ISSC), pp. 1–4. doi:10.1109/ISSC.2018.8585340.

Duda, R.O., Hart, P.E., 1972. Use of the hough transformation to detect lines and curves in pictures. Communications of the ACM 15, 11–15.

Girshick, R., 2015. Fast r-cnn doi:10.1109/ICCV.2015.169.

Girshick, R., Donahue, J., Darrell, T., Malik, J., 2013. Rich feature hierarchies for accurate object detection and semantic segmentation. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition doi:10.1109/CVPR.2014.81.

Grini, S.V., Brekke, E., 2019. Object detection in maritime environments. NNU Norwegian University of Science and Technology, Faculty of Information Technology and Electrical Engineering, Department of Engineering Cybernetics .

Gul, S., Shiriyev, J., Singhal, V., Erge, O., Temizel, C., 2021. Advanced materials and sensors in well logging, drilling, and completion operations. 1 ed., Gulf Professional Publishing. doi:10.1016/B978-0-12-824380-0.00004-9.

Helgesen, K., Brekke, E.F., Stahl, A., Engelhardtsen, , 2020. Low altitude georeferencing for imaging sensors in maritime tracking. IFAC-PapersOnLine 53, 14476–14481.

Hulstaert, L., 2018. A beginner's guide to object detection. URL: https://www.datacamp.com/community/tutorials/object-detection-guide.

Kwiatkowski, R., 2021. Gradient descent algorithm — a deep dive. URL: https://towardsdatascience.com/gradient-descent-algorithm-a-deep-dive-cf04e8115f21.

Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollar, P., 2017. Focal loss for dense object detection, pp. 2999–3007. doi:10.1109/ICCV.2017.324.

MaritimeRobotics, 2022. Otter. URL: https://www.maritimerobotics.com/otter.

Padilla, R., Netto, S., da Silva, E., 2020. A survey on performance metrics for object-detection algorithms. doi:10.1109/IWSSIP48289.2020.

Park, J., Kim, J., Son, N.s., 2015. Passive target tracking of marine traffic ships using onboard monocular camera for unmanned surface vessel. Electronics Letters 51. doi:10.1049/el.2015.1163.

Pinggera, P., Franke, U., Mester, R., 2015. High-performance long range obstacle detection using stereo vision. 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) , 1308–1313.

Redmon, J., Divvala, S., Girshick, R., Farhadi, A., 2016. You only look once: Unified, real-time object detection, pp. 779–788. doi:10.1109/CVPR.2016.91.

Redmon, J., Farhadi, A., 2017. Yolo9000: Better, faster, stronger, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 6517–6525. doi:10.1109/CVPR.2017.690.

Redmon, J., Farhadi, A., 2018. Yolov3: An incremental improvement. CoRR abs/1804.02767. URL: http://arxiv.org/abs/1804.02767, arXiv:1804.02767.

Ren, S., He, K., Girshick, R., Sun, J., 2017. Faster r-cnn: Towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis and Machine Intelligence 39, 1137–1149. doi:10.1109/TPAMI.2016.2577031.

Rosten, E., Porter, R., Drummond, T., 2010. Faster and better: A machine learning approach to corner detection. IEEE Transactions on Pattern Analysis and Machine Intelligence 32, 105–119. doi:10.1109/TPAMI.2008.275.

Sabbatelli, M., Bernini, N., Bertozzi, M., Castangia, L., Patander, M., 2014. Realtime obstacle detection using stereo vision for autonomous ground vehicles: A survey. doi:10.1109/ITSC.2014.6957799.

Saha, S., 2018. A comprehensive guide to convolutional neural networks — the eli5 way. URL: https://towardsdatascience.com/ a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3

Sandler, M., Howard, A.G., Zhu, M., Zhmoginov, A., Chen, L., 2018. Inverted residuals and linear bottlenecks: Mobile networks for classification, detection and segmentation. CoRR abs/1801.04381. URL: http://arxiv.org/abs/1801.04381, arXiv:1801.04381.

Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., Lecun, Y., 2013. Overfeat: Integrated recognition, localization and detection using convolutional networks. International Conference on Learning Representations (ICLR) (Banff) .

Talukder, A., Manduchi, R., Rankin, A., Matthies, L., 2002. Fast and reliable obstacle detection and segmentation for cross-country navigation. IEEE Intelligent Vehicle Symposium .

Wei Liu, Dragomir Anguelov, D.E.C.S.S.R.C.Y.F.A.C.B., 2016. Ssd: Single shot multibox detector. Computer Vision – ECCV 2016. ECCV 2016. Lecture Notes in Computer Science 9905, 21–37.

NTNU

Norwegian University of
Science and Technology