UNIVERSITY OF TARTU
Faculty of Science and Technology
Institute of Technology
Robotics and Computer Engineering Curriculum

Fidan Rustambayli

Comparison of Water Detection Models for an Off-road Unmanned Ground Vehicle

Master's Thesis (30 ECTS)

Supervisor(s): Joosep Kivastik
Mihkel Pajusalu

Tartu 2023

Comparison of Water Detection Models for an Off-road Unmanned Ground Vehicle

Abstract: Water hazards can cause unmanned ground vehicles (UGVs) to become stuck or break down during an autonomous mission, damage electronic components and sensors, and require costly repairs or replacements, making it crucial for UGVs to identify water hazards in real-time, determine secure path around them, or reduce their speed when appropriate to cross them safely. This thesis proposes a water detection system for UGVs in off-road environment. The proposed approach combines convolutional neural networks (CNNs) with transfer learning, leveraging their capabilities for effective water detection. The thesis includes a comprehensive review of traditional sensor-based methods and recent deep learning-based techniques. Real-world data collected in off-road environments are utilized to evaluate the proposed approach, and the method achieves a 0.50 Mean-IoU score and 92.74% accuracy on the test dataset. We also include a comparative analysis of the method with a previous deep learning-based semantic segmentation method for water detection. The comparison provides insights into the relative strengths and weaknesses of these approaches for water detection in off-road environments. Overall, this thesis provides valuable insights into the use of deep learning for semantic segmentation in challenging environments.

Keywords:
Water detection, deep learning, transfer learning, object detection, convolutional neural networks (CNNs), Unmanned ground vehicles (UGVs), Off-road environments

CERCS: T125 - Automation, robotics, control engineering, P170 - Computer science, numerical analysis, systems, control, P176 - Artificial intelligence, T111 Imaging, image processing

Veedetektorite mudelite võrdlus maastikul sõitva mehitamata maismaasõiduki jaoks

Resümee Veetakistused võivad põhjustada mehitamata maasõidukite (UGV-de) kinnikiilumise või katkemise autonoomse missiooni ajal, kahjustada elektroonilisi komponente ja sensoreid ning nõuda kulukaid parandusi või asendusi, muutes oluliseks UGV-de võime tuvastada veeohud reaalajas, määrata ohutu tee nende ümber või vajadusel vähendada kiirust, et neid ohutult ületada. See väitekiri pakub välja veetuvastussüsteemi UGV-dele maastikul. Pakutud lähenemisviis ühendab konvolutsioonilised tehisnärvivõrgud (CNN-id) ülekande õppega, kasutades ära nende võimeid efektiivseks veetuvastuseks. Väitekiri hõlmab põhjalikku ülevaadet traditsioonilistest sensoripõhistest meetoditest ja hiljutistest sügavõppe-põhistest tehnikatest. Reaalse maailma andmeid, mis on kogutud maastikul,

kasutatakse pakutud lähenemisviisi hindamiseks. Meetod saavutab testandmestikus 0,50 keskmise IoU skoori ja 92,74% täpsuse. Lisaks analüüsime meetodit eelmise sügavõppepõhise meetodiga. Võrdlus annab ülevaate nende lähenemisviiside suhtelisest tugevusest ja nõrkustest veetuvastuses maastikel. Kokkuvõttes pakub see väitekiri väärtuslikke teadmisi sügavõppe semantilise segmentatsiooni kasutamisest keerukates keskkondades.

Märksõnad: vee tuvastamine, süvaõpe, ülekande õpe, objekti tuvastus, konvolutsioonilised närvivõrgud, mehitamata maismaasõidukid, maastiku keskkond

CERCS: T125 - Automatiseerimine, robootika, control engineering, P170 - Arvutiteadus, arvutusmeetodid, süsteemid, juhtimine, P176 - Tehisintellekt, T111 - Pilditehnika

Acknowledgment

Our developments are officially for robotic forestry and the work is part of project "Rakendusuuring metsauuendussüsteemi Robotic Forester juhtimisautonoomia suurendamiseks (Nutikas Metsarobot)“, which was supported by European Union European Regional Fund. I would like to express my gratitude to all the team members for their exceptional collaboration and remarkable teamwork in achieving our goals.

Fidan Rustambayli

European Union
European Regional
Development Fund

Investing
in your future

Contents

List of Tables

List of Figures

1 Introduction

Unmanned ground vehicles (UGVs) are increasingly being used in various applications, including military operations [5], surveillance, logistics, and rescue operations [6, 7, 8] in the last decades. However, UGVs face many challenges when navigating over complex terrain, including the presence of various water sources such as stagnant water, puddles, lakes, and ponds during operations. Water detection is a critical perception requirement for UGVs to navigate autonomously and avoid hazards, such as vehicle immobilization. In this thesis, we focus on off-road water detection for UGVs using deep learning. Our objective is to develop an accurate water detection system through extensive research and comparison of relevant works, particularly focusing on deep learning methods. By leveraging the advancements in deep learning techniques, we aimed to integrate this system into the existing UGV navigation system, thereby enhancing safety and reliability in challenging environments.

1.1 Background and motivation

Ground robots, whether for military or civilian applications, were increasingly popular in the second half of the twentieth century. During this time, ground robots with integrated sensors and remote control capabilities were developed, allowing them to achieve autonomous features while requiring less operator input [5]. Unmanned ground vehicles have been used in military operations for decades to conduct duties such as transporting supplies [9, 10], and clearing minefields [11]. They have also been used for reconnaissance and surveillance activities such as exploring and obtaining information about the battlefield. The increased need for a combat-focused UGV, particularly in military situations, has corresponded to modern power competition. Military technology advances as the nature of battle evolves. [5]

Military unmanned ground vehicles are required to operate in more harsh environments than civilian robots due to the nature of their tasks. An increase in any of the three primary types of complexity (mechanical, environmental, and mission or task complexity) radically increases the required capabilities of a robot. This creates challenges in developing UGVs that are capable of navigating various terrains and effectively avoiding obstacles, especially in military applications. [12]

One of the challenges in defense land-robotic research is the ability to navigate various terrains, such as uneven, bumpy, slippery, rocky, icy, and others, and effectively avoid obstacles. Military robots require autonomous features to enable them to navigate, make decisions, and respond to their surroundings. [13] It is a fundamental task for a ground robot to perceive its environment and stop or maneuver when encountering any hazard.

Water bodies can present significant challenges for UGVs due to the potential for costly damage to their electronics when navigating deep water or becoming stuck. In

addition, if a UGV breaks down or becomes stuck in water during an autonomous mission, it may require rescue, which can divert critical resources from the primary mission and put soldiers in harm's way [13]. Therefore, it is crucial for UGVs to identify water hazards in real-time, determine a secure path around them, or reduce their speed when appropriate to cross them safely. It is essential to have a precise and rapid detection system that can provide immediate assistance or path planning. Despite advances in UGV technology, water detection remains a challenging problem due to the variability of water sources, weather conditions, and terrain features. Existing water detection techniques often rely on specialized sensors [14, 15, 16, 17], which can be expensive, time-consuming, or impractical in certain situations. Deep-learning techniques offer a promising solution to this problem, as they can learn to recognize water features from camera images and adapt to changing conditions.

1.2 Related Work

Numerous research studies have utilized various sensor technologies to detect water across various applications. Our research methodology involved a comprehensive search of academic databases such as Google Scholar and Scopus using predefined search terms such as "off-road water detection," "water hazard detection," "puddle detection," and "water detection with deep learning" for relevant publications since 2012. This enabled us to gather a wide range of research studies that have employed different sensor technologies for water detection in various applications. These studies can be classified into two categories: those using deep learning and those using other techniques. In this chapter, we have summarized them to highlight the limitations and to identify significant challenges associated with water detection.

1.2.1 Other technologies

Various studies have utilized stereo cameras and visions to detect water hazards. [17], [18], [16], [19] One study presents an approach for detecting traversable areas and water hazards using a polarized RGB-Depth (pRGB-D) sensors, which can assist visually impaired people in navigating their environment safely. The approach involves enhancing a stereo camera with an attitude angle sensor and horizontal/vertical polarizers to create a pRGB-D sensor. By combining polarization-color-depth-attitude information the system generates a point cloud in 3D space. The effectiveness and robustness of the approach are demonstrated through experiments and a user study. The results show that the proposed approach can achieve a high detection rate while maintaining low false positives. A limitation of this technology is that it relies on a stable attitude angle of the sensor to produce precise detection outcomes, which can be challenging to maintain in certain situations. Additionally, the approach may not perform well when applied to highly reflective or transparent water surfaces. [16] Subsequently, a new study was published by the same authors of the paper. The study introduced a framework incorporating polarization imaging, RGB-D sensor awareness, and real-time semantic segmentation to detect water hazards beyond traversability. The outcomes of this study demonstrated that the system had improved speed, efficiency, and robustness when used in various perceptual and environmental conditions [17].

A system was created to identify water hazards on roads by analyzing variations in the polarization and color of reflected light in the paper [18]. A stereo-polarization camera system was used to capture images from two different angles and polarizations. By analyzing the differences in polarization and color between the two images, the system can identify areas where there is water on the road. The study examined how the color and polarization of light reflected off water surfaces change depending on factors such as the angle of incidence and direction of observation. This approach allows for reliable water detection and tracking over a wide range of realistic car driving water conditions

using polarized vision as the primary sensing modality. The system successfully detects water hazards up to more than 100m. [18]

The authors (Kim et al. 2016) propose a new method for detecting wet areas and puddles on the road using a stereo camera. The proposed method has two stages: hypothesis generation and hypothesis verification. In the hypothesis generation stage, color information is used to generate a hypothesis about the presence of wet areas or puddles. In the hypothesis verification stage, a support vector machine algorithm is used to verify the hypothesis generated in the previous stage using three features: polarization difference, graininess, and gradient magnitude. If the support vector machine classifies the input feature vector as a wet area or puddle, it is considered a positive detection. The system shows promising results in detecting water areas. However, The resulting images indicate that the segmented areas have noticeable pixelation. [19]

The authors (Billah et al. 2021) propose a spatiotemporal approach for identifying water regions in videos, which involves using a combination of texture, color, and motion features to recognize and segment water in different environments.

- Preprocessing: Grayscale pictures are created from the video frames after they have been filtered to remove noise.
- Feature extraction: Local Binary Patterns are used to extract texture characteristics, color histograms are used to extract color information and optical flow calculation is used to extract motion features.
- Classification: A Support Vector Machine classifier is used to classify each pixel as either water or non-water based on its feature vector.
- Segmentation: The method divides the preprocessed video into two segments, classifies potential locations as either having water or not, and then applies a second segmentation to resegment the result.

In some environments, there is a risk of false detections because water areas may appear visually similar to other objects or regions. Additionally, this method may not be effective at detecting small or partially hidden water areas. [20]

The proposed method for water detection in the paper [21] incorporates texture, color, and reflections as integral components of its multi-cue approach. The software first examines a specific area's color to see if it resembles the blue or green colors that are typically associated with water. The texture of the area is then examined to see if it resembles water, which has a generally smooth texture. Lastly, the software searches for regions where light reflects off the surface at different angles to detect reflections, which are typical properties of water. [21]

In another work [14], a 2D Laser Range Finder sensor is used for detecting and avoiding puddles on narrow roads. The sensor is mounted on an autonomous mobile

robot. The method measures the reflection intensity of the road surface using the 2D Laser Range Finder sensor and analyzes these measurements to detect puddles. The technique detects the presence of puddles by comparing the reflection intensities from the water surface of a puddle and the road surface. The effectiveness of this method was evaluated by measuring the success rate of detecting and avoiding puddles in different road conditions. The results showed that the proposed method was successful in detecting and avoiding puddles, even in challenging road conditions such as those with varying reflection intensities. [14].

The paper [22] proposes a method for detecting water in video sequences by recognizing the characteristic pattern of water through segmentation-guided dynamic texture analysis. The technique involves computing an entropy measure from the optical flow across multiple frames to identify the dynamic texture of the water in the video sequences. Subsequently, The proposed method uses a segmentation-guided label propagation method to expand the water detection results to motionless regions of the input image. This means that after detecting water in the video sequence using the entropy measure computed from optical flow, the method further processes the image to identify other regions that are likely to be water but were not detected by the initial detection process. Several video experiments were performed to validate the method's effectiveness. However, it should be noted that the method assumes water to have a chaotic dynamic texture, which may not hold in all real-world scenarios where the water is stagnant. [22]

The proposed method in the paper [23] uses a set of features called spatio-temporal invariant descriptors to detect water in videos. These descriptors capture the motion characteristics of water and are used to classify the presence of water in each local region. In simpler terms, this method looks for specific patterns that are common in water motion and uses them to identify where there is water in a video. The proposed method for water detection in videos has some limitations. For example, it may not accurately distinguish between different water bodies that have similar local textures but different global aspects, such as shape or size. Additionally, the algorithm's performance may be compromised in cases where the water is not clearly distinguishable from the background or has a complex texture [23].

The authors propose (Shao et al. 2021) a mathematical model of the line-structured light sensor that was established to measure water obstacles in long-range and all-day conditions, analyzing how the laser beam interacts with water and how reflected light can determine the obstacle's depth and shape. Next, a new method was developed to extract the contour of water hazards using the sensor's characteristics, analyzing how the laser stripe changes when encountering water and other obstacles. A filtering method was also proposed to reduce noise in collected sensor data. A series of experiments were conducted to test the sensor, which demonstrated high accuracy and speed in detecting water hazards from long distances. When the laser beam encounters water, most of the light is reflected and rarely refracted due to the angle between the light and the ground

being relatively small. This makes it difficult to detect the depth of the water. The sensor is suitable for small water bodies such as puddles but not for large ones such as ponds and lakes [24].

Using a camera for perception tasks on a UGV is a cost-efficient solution compared to other sensor technologies. While other sensors like Light Detection and Ranging (LiDAR) and radar have their advantages, utilizing an already present camera on the ground vehicle makes it a more practical option. Additionally, the camera can perform other object detection tasks, making it a versatile solution. In the case of water detection, the camera can be utilized for overall traversable area [25, 26] detection and for detecting the hazards on the road [27]. Overall, the use of the existing camera on the UGV is a beneficial choice, especially when the cost is a significant factor in the decision-making process.

1.2.2 Deep Learning

Deep learning has gained significant attention in recent years due to its high accuracy in various applications in object detection [28, 29, 30]. Deep learning models are trained on large amounts of data and can learn complex patterns and features, which makes them well-suited for water detection tasks. Deep learning techniques have shown promising results in water detection studies [4, 31, 32], outperforming traditional methods such as thresholding [21] or image processing algorithms [23].

The paper presents a method for detecting water puddles on and off the road using a Fully Convolutional Network (FCN) with Reflection Attention Units. According to the study, the dynamic texture of water surfaces, reflections from the sky and nearby scenes, and changes in lighting conditions pose the primary challenge in detecting water from a moving camera. To overcome these challenges, the method utilizes Reflection Attention Units, which are specifically designed to capture the reflection of water with its surroundings in distinct vertical sections of images. The authors of the paper gathered a dataset of color stereo images with polarizers to train and test their method. [4]. To evaluate the effectiveness of the proposed method for water detection, we conducted experiments and compared its results with those obtained by other methodologies.

The deep-learning-based water detection network proposed in the paper exploits temporal information across 8 continuous frames to detect water at an accuracy approaching the state of the art. [31] The authors (Li et al. 2019) proposed another study with a slightly modified version of the above-mentioned network. T3D-FCN network architecture has an additional pooling layer after the second convolutional block [32].

The study conducted by the authors (Tas et al. 2021) aims to address the challenge of water detection in low-power microcomputers by proposing a solution that utilizes the power of cloud systems. The methodology involves the use of cloud-based deep learning techniques for real-time object detection. The approach leverages a YOLO-based system and convolutional neural network to accurately detect water in images. [33]

The methodology of the paper [34] involves a novel automatic stagnant water localization method under weak supervision based on visual images. The authors (Zhao et al. 2022) first apply the template matching method to extract road information from the traffic image. Then, they locate stagnant water in the image based on Class Activation Maps (CAM) mechanism, which is a weakly supervised method. The detection model consists of the ResNet-18 and Grad-CAM++ mechanisms. Finally, based on the heat map and template, they set a suitable threshold to segment the stagnant water area in the image. The proposed methodology is designed to improve accuracy and generalization ability in detecting stagnant water in road traffic images. It appears that the proposed method is specifically tailored for detecting still water bodies and may not be suitable for identifying dynamic or turbulent water conditions. Furthermore, the methodology for road segmentation relies on the use of static cameras, a condition that is not applicable to the context of UGVs which are designed to traverse through variable terrains.

1.3 Research question and contributions

In this thesis, we aim to answer the following research question: Can CNN models with transfer learning outperform existing deep learning techniques in developing an accurate water detection system for seamless integration into the UGV navigation system, thereby improving safety and reliability in challenging off-road environments? Our contributions include a comprehensive literature review of existing water detection methods, an evaluation of different CNN architectures with transfer learning and training strategies, and a validation of our proposed approach using real-world data.

Specifically, we investigate the effectiveness of pre-trained segmentation models that are fine-tuned for our water detection task. We have also applied data augmentation to increase the diversity and quantity of our training data. To evaluate the performance of our approach, we conduct experiments on a publicly available dataset as well as on a custom dataset collected using a UGV equipped with various sensors.

Overall, this thesis contributes to the field of robotics by providing a practical solution to an important perception problem for UGVs. Our work has potential applications in agriculture mining industries where off-road vehicles require water detection capabilities to navigate challenging environments.

1.4 Thesis outline

The thesis is structured as follows: Chapter 1 provides an introduction to the background, problem statement, research question, and contributions of this thesis. We also present a literature review of traditional water detection methods and deep learning-based approaches in this chapter. Chapter 2 describes the methodology used in this study, including dataset preparation and augmentation, CNN architectures and transfer learning techniques, training strategies, and hyperparameter tuning. Chapter 3 presents the experimental results and analysis of our proposed approach. Finally, Chapter 4 and 5 concludes the thesis by summarizing our contributions, and discussing limitations and future directions for research in this area.

2 Deep Learning

Deep learning is a subfield of machine learning that resembles the information-processing mechanism of the human brain. The processing mechanisms employed by humans, such as vision and hearing, indicate that deep architectures are necessary to extract intricate structures and construct internal representations from extensive sensory inputs [35]. Hence, Deep learning is experiencing notable progress in addressing challenges that have been particularly challenging for the artificial intelligence community over the years [36]. Deep learning's primary benefit is that it can learn useful features using a general learning process, eliminating the need for handcrafted feature extractors that require significant engineering skills and domain expertise. [36]

Deep learning has been very successful in solving many complex problems, such as computer vision and image recognition [37, 38, 39], speech recognition [40, 41] and natural language processing [42, 43], [35]. It has enabled significant advances in fields such as robotics, and autonomous vehicles [44, 45].

2.1 Neural Networks

Deep learning uses neural networks to learn complex data representations with various levels of abstraction [36]. Neural networks (Figure 1) are made up of node layers that include an input layer, one or more hidden layers, and an output layer. Nodes in the network cascade data from one layer to the next, resulting in increasingly complex decision-making. Every artificial neuron or node is connected to another, and they are assigned a weight and a threshold. When the output of a node exceeds the threshold, it becomes activated and passes data to the next layer of the network. Otherwise, no data is passed along.[46]

The majority of deep neural networks follow a feedforward architecture, where information travels only in one direction, from the input layer to the output layer. However, it is possible to utilize backpropagation to train the model in reverse, moving from the output to the input layer. Backpropagation enables us to compute and assign the error associated with each neuron, enabling us to fine-tune and optimize the model parameters. [46]

The most common types of neural networks include perceptrons, feedforward neural networks (multi-layer perceptrons), convolutional neural networks, and recurrent neural networks. Feedforward neural networks are the foundation for computer vision, natural language processing, and other neural networks, while convolutional neural networks are usually utilized for image recognition and pattern recognition. [46]

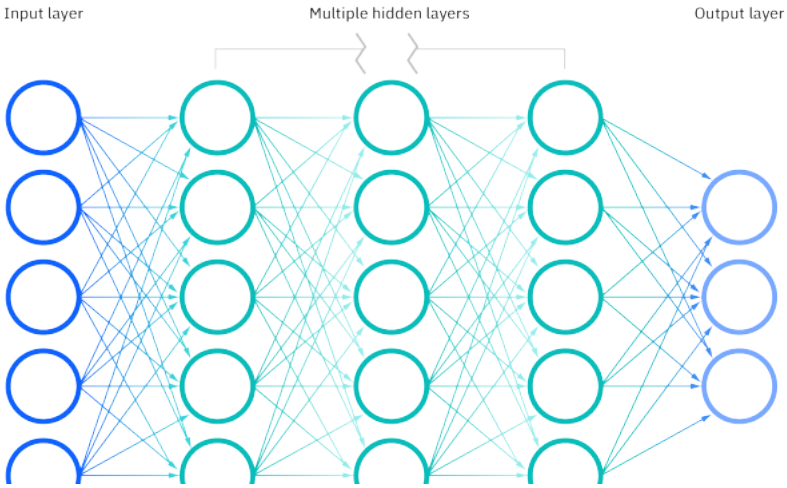


Figure 1. Neural network: input layer, hidden layers, and output layers. [1]

2.2 Supervised Learning

Supervised learning is a type of machine learning that uses labeled datasets to train algorithms to accurately classify data or predict outcomes. The model adjusts its weights until it is appropriately fitted during cross-validation. The training set includes inputs and correct outputs, allowing the model to learn over time and minimize errors through the loss function.[36]

2.3 Convolutional neural networks

Convolutional neural networks are a type of neural network that performs well for image and audio data. CNNs have three main types of layers, including convolutional, pooling, and fully-connected layers. [47]

2.3.1 Convolutional Layer

The convolutional layer is a crucial part of a CNN where most of the computation occurs. In the convolutional layer, a feature detector, also known as a kernel or filter, is used to convolve the input with the kernel. The filter, typically a 3x3 matrix, is applied to an area of the image, and a dot product is calculated between the input pixels and the filter to generate a feature map. This process is repeated across the entire image, and a Rectified Linear Unit transformation is applied to introduce nonlinearity to the model. Another convolution layer can follow the initial convolution layer to create a hierarchical structure in the CNN, with lower-level patterns in the image being combined to represent higher-level patterns. [47]

2.3.2 Pooling Layer

Pooling layers in CNNs perform dimensionality reduction by reducing the number of parameters in the input. They use a filter to apply an aggregation function to the values within a receptive field, populating the output array. There are two types of pooling: max pooling and average pooling. Max pooling selects the pixel with the maximum value, while average pooling calculates the average value within the receptive field. Although some information is lost during pooling, it improves efficiency and limits overfitting in CNNs. [47]

2.3.3 Fully-Connected Layer

The fully-connected layer is responsible for classification based on the features extracted from previous layers of the CNNs. Unlike convolutional and pooling layers, it connects each node in the output layer directly to a node in the previous layer. The layer uses a softmax activation function to classify inputs, producing a probability from 0 to 1. [47]

2.4 Deep Learning in Object Detection

Due to the advancements in Deep Convolutional Neural Networks (DCNNs) and the increased computing capabilities of GPUs, deep learning models are now widely utilized in the field of computer vision [48, 49]. The primary objectives of object detection are to identify and classify objects, which are achieved by utilizing rectangular bounding boxes. Object classification, semantic segmentation, and instance segmentation (Figure 2) are all associated with object detection to some extent. [50]

Bounding box is the simplest method and involves drawing a rectangular box around an object in an image. It is mostly used in tasks where the focus is on locating the object of interest rather than precisely identifying its boundaries. [49]

Semantic segmentation is a sophisticated approach that involves labeling each pixel in an image to a corresponding class. This method is particularly beneficial for tasks that require identifying the precise object boundaries and distinguishing them from other class objects in the image. [49]

Instance segmentation is an advanced technique that not only identifies and classifies each object's pixels in an image but also accurately delineates object boundaries within the same class. This technique is particularly useful for tasks that require precise detection and identification of individual objects in an image. [49]

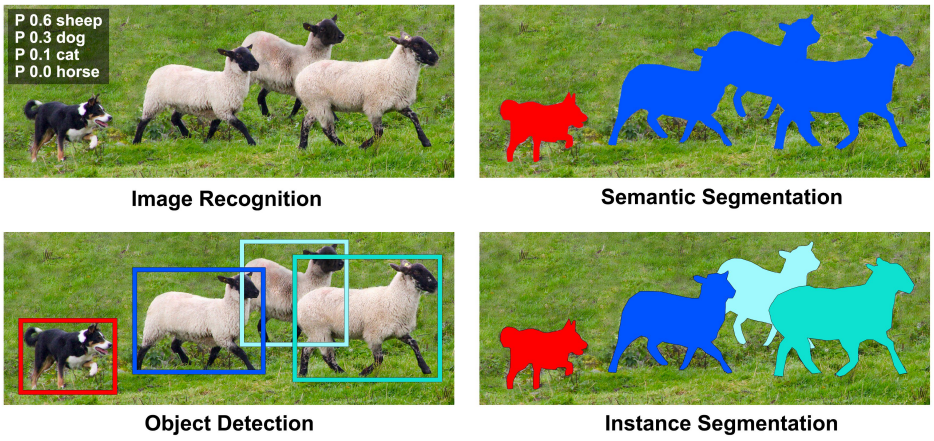


Figure 2. An example image from the COCO dataset is provided to demonstrate the distinction between image-level annotations, object-level annotations, and segmentations at the class/semantic- or instance level. [2]

Object detection has numerous applications in the field of transportation and medical sciences. In transportation, object detection is used for traffic monitoring, pedestrian detection, and vehicle detection. In medical science, it is used for detecting tumors in medical images, identifying anatomical structures, and tracking cells in live-cell imaging. Object detection requires high accuracy and precision, as well as the ability to handle

complex scenes with multiple objects of different sizes, shapes, and orientations. [49] DCNNs use multiple layers of convolutional filters to extract increasingly complex features from the input image. The first few layers typically learn low-level features such as edges and corners, while later layers learn higher-level features such as object parts and textures. By combining these learned features, the network can make accurate predictions about the presence and location of objects in an image. [50]

2.4.1 Backbone networks CNN

Backbone architectures are the fundamental building blocks of deep learning models used for object detection. They are responsible for extracting feature maps from the input image, which are then used to detect and classify objects. There are several backbone architectures used in object detection models, such as VGGNet [51], ResNet [52], and AlexNet [53]. Each architecture has its unique characteristics and performance trade-offs. [49]

2.4.2 Benchmark datasets

Benchmark datasets are standardized datasets that are widely used to evaluate the performance of models. Some of these datasets contain a large number of images with annotated objects, which are used to train and test object detection models. Benchmark datasets are essential for comparing the performance of different models and for tracking progress in the field. They provide a common ground for researchers to evaluate their models and compare them with state-of-the-art methods. Some popular benchmark datasets used in object detection research include PASCAL VOC [54], MS COCO [55], ImageNet [56], KITTI [57], and SUN [58]. These datasets vary in size, complexity, and annotation quality, which makes them suitable for evaluating different aspects of object detection models. [49]

2.4.3 Evaluation metrics

Evaluation criteria are the metrics used to measure the performance of object detection models on benchmark datasets or custom datasets. These metrics are used to evaluate the accuracy, precision, recall, and other aspects of object detection models. [49]

Variables used for the calculation of metrics are:

- *TP (True Positive) - Model has predicted positive and in actual it's true.*
- *TN (True Negative) - Model has predicted negative and in actual it's true.*
- *FP (False Positive) - Model has predicted positive and in actual it's false.*
- *FN (False Negative) - Model has predicted negative and in actual it's false.*

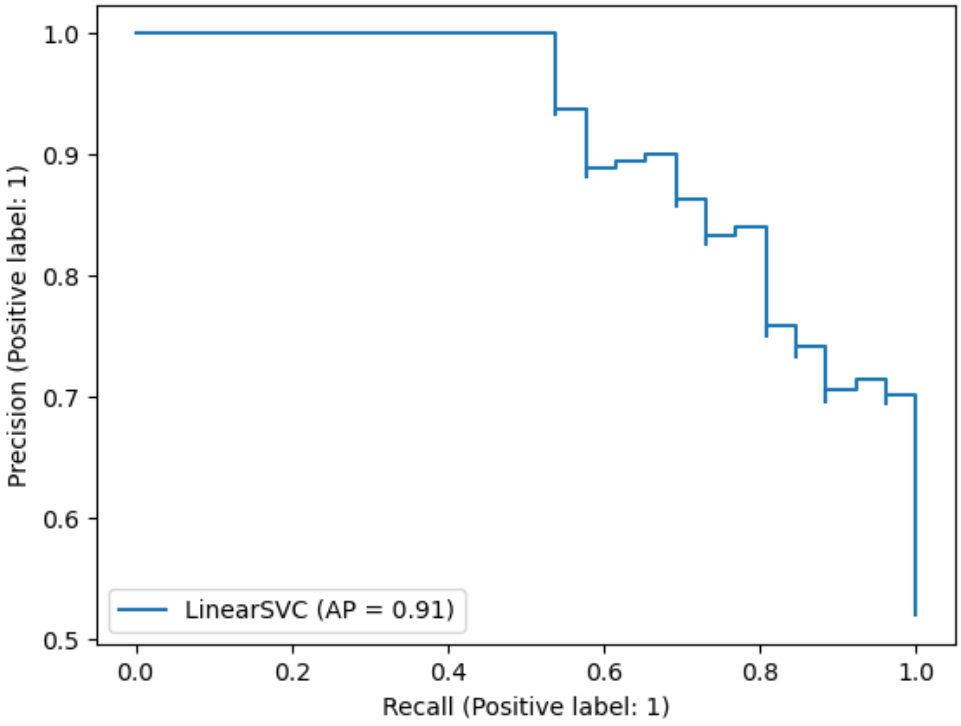


Figure 3. Precision-Recall curve in 2 class classification [1]

- $C_{mn} = C_m$ *is the category in the instance or image.*
- i = no. of instances in a class.
- j = no. of categories.
- BB = bounding box

$$Accuracy_{C_{mn}} = \frac{TP_{C_{mn}} + TN_{C_{mn}}}{TP_{C_{mn}} + FP_{C_{mn}} + TN_{C_{mn}} + FN_{C_{mn}}} \tag{1}$$

$$Precision_{C_{mn}} = \frac{TP_{C_{mn}}}{TP_{C_{mn}} + FP_{C_{mn}}} \tag{2}$$

$$Recall_{C_{mn}} = \frac{TP_{C_{mn}}}{TP_{C_{mn}} + FN_{C_{mn}}} \tag{3}$$

Precision-recall (PR) curve (Figure 3) is a graphical representation of the trade-off between precision and recall for different threshold values. It is used to evaluate how well an object detector can detect objects at different levels of confidence. [49]

Some common evaluation criteria used in object detection research include mean average precision (mAP), intersection over union (IoU), precision-recall (PR) curve, and F1 score. [49]

Mean average precision (mAP) is a widely used metric that measures the average precision of an object detector across different levels of recall. It is calculated by averaging the precision values at different recall levels. [49]

$$AP_{C_m} = \frac{1}{j} \sum_{m=1}^{j} Precision_{C_{mn}} \tag{4}$$

$$mAP = \frac{1}{j} \sum_{m=1}^{j} \frac{AP_{C_m}}{n} \tag{5}$$

Intersection over union (IoU) measures the overlap between the predicted bounding box and the ground truth bounding box. It is calculated as the ratio of intersection area to union area between two bounding boxes. [49]

$$\begin{aligned} IoU &= J(BBpredict, BBground) \\ &= \frac{\textit{Area of intersection of predicted and ground truth boxes}}{\textit{Area of union of predicted and ground truth boxes}} \end{aligned} \tag{6}$$

F1 score is a weighted average of precision and recall that balances both metrics equally. It is often used as a single metric to evaluate the overall performance of an object detector [49]

$$\textit{F1-score} = 2 * \frac{\textit{Precision} * \textit{Recall}}{\textit{Precision} + \textit{Recall}} \tag{7}$$

2.5 ResNet

ResNet [52] is a deep learning model developed by He et al. in 2016 to address the problem of training deep neural networks. ResNet uses residual connections that enable the network to skip some layers and directly propagate information from one layer to another, which makes it easier to optimize the network and achieve higher accuracy. ResNet outperformed other models in image classification, indicating that it extracts image features well [59].

It has less computational complexity than previous architectures like AlexNet [53] and VGGNet [51]. ResNet50 [52] and ResNet101 [52], with skip connections to preserve gradients in deeper layers, are commonly used. ResNet101 performs similarly to VGG network but with fewer parameters, using global average pooling and bottleneck like GoogLeNet [60]. [49]

2.6 Mask R-CNN

Mask R-CNN [61] is an object detector designed to solve the instance segmentation issue by performing pixel-level segmentation. Mask R-CNN is an augmentation of Faster R-CNN [62] and follows its architecture but has three outputs for each object proposal, including class label, bounding box offset, and object detection mask. The RoIAlign layer is used in Mask R-CNN to associate extracted features with the object's input position and fix misalignment issues in the RoI pooling layer. By using bilinear interpolation, the RoIAlign layer evaluates the real feature values at each sampling point and eliminates the need to measure the RoI threshold. Mask R-CNN achieved state-of-the-art performance on instance segmentation. [49]

2.7 DeepLab

DeepLab [63] is a popular deep learning architecture for semantic segmentation that has shown impressive performance on various datasets. The method can efficiently create detailed segmentation maps and make semantically accurate predictions by utilizing fully connected conditional random fields (CRF) [64].

DeepLabv1 (DeepLab [63]): This network uses atrous convolution [63] to control the resolution of feature responses within Deep Convolutional Neural Networks. Atrous convolution is a type of convolutional operation used in deep learning for image analysis. It is also known as dilated convolution. [63]

DeepLabv2 [63]: To segment objects at multiple scales, DeepLabv2 employs atrous spatial pyramid pooling (ASPP) [63], which uses filters at different sampling rates and effective fields-of-views.

DeepLabv3 [65]: DeepLabv3 enhances the ASPP module with image-level features to capture longer-range information and includes batch normalization parameters to facilitate training. The network uses atrous convolution during both training and evaluation to extract output features at different output strides, resulting in efficient training of BN at output stride = 16 and high performance at output stride = 8 during evaluation.

DeepLabv3+: In addition to the ASPP module, DeepLabv3+ includes a decoder module that refines segmentation results, especially along object boundaries. The network also uses atrous convolution to control the resolution of extracted encoder features, allowing for a trade-off between precision and runtime. [66]

2.8 Transfer learning

Deep learning models often struggle when there is a limited amount of data available, which can make it difficult or even impossible to obtain sufficient datasets for practical applications. The annotation process can be laborious and time-consuming, and outsourcing it may also come at a cost. However, techniques such as transfer learning have been

developed to address this issue. [67]

Transfer learning involves using a pre-trained network from a source domain and task where a large dataset is available, and adapting it for use in the target domain and task, which is similar to the original task and domain [68]. [67]

3 Methodology

The aim is to achieve pixel-wise accuracy in identifying water regions within the UGV's environment for off-road UGVs using deep learning-based semantic segmentation. We focused on exploring and evaluating the most recent academic papers and approaches related to water detection using semantic segmentation networks in this study. Furthermore, an important objective of this research is to test and compare the performance of multiple pre-trained semantic segmentation deep learning models in the context of water detection.

3.1 Dataset Preparation

In the initial stage, we, the project team, gathered an off-road dataset (Figure 4) specifically composed of water bodies such as puddles, lakes, rivers, and seashores. We annotated the dataset by segmenting water bodies in the images and creating a binary segmentation mask. The dataset was obtained from more than 60 videos recorded using a camera mounted on a UGV, resulting in the presence of repetitive frames. To ensure a well-balanced dataset, we carefully filtered out the redundant frames when splitting the data into distinct test and validation datasets. It is important to note that the dataset includes a relatively smaller number of puddle images, with a primary focus on lakes, seas, seashores, and water bodies in dense vegetation.

In addition to our custom dataset, we incorporated the Atlantis dataset (Figure 5) [3] into our research. This dataset encompasses a wide range of water bodies, including lakes, rivers, seas, puddles, canals, and more. The original intention behind this dataset was to gather data for a classification task, the dataset also includes labeled objects surrounding the water bodies, such as humans and other objects. However, for our specific study, we focused solely on extracting and generating binary masks for the water bodies, discarding the labels for other objects.

Additionally, we incorporated the Puddle-1000 [4] dataset (Figure 6) into our research, which consists of a collection of video frames specifically focused on puddles. This dataset encompasses water hazards present in on-road and off-road environments.

To ensure a real-world assessment of the networks, we utilized exclusively our custom dataset in the test and validation phase. This decision was made as the dataset was captured using a UGV camera, providing a representative depiction of real-world scenarios. The size of the training dataset is approximately 8000, while the validation and test datasets are approximately 500 each.

(a) puddle (b) seashore

(c) water in dense vegetation (d) seashore

Figure 4. Custom dataset captured with a UGV camera

(a) Hot spring

(b) Sea

(c) Puddle

(d) Flood

Figure 5. Atlantis dataset [3]

(a) Off-road water hazard

(b) On-road water hazard

Figure 6. Puddle-1000 dataset [4]

3.2 Experimental Setup

We conducted our experiments using the following software setup:

- Programming Language: Python 3.10.8 [69]
- Deep Learning Framework: PyTorch [70]: A popular deep learning framework that provides tools and functionalities for building and training neural networks.

To implement our water detection system and perform the experiments, we utilized the following specific libraries and modules:

- Torchvision [71]: A PyTorch library that provides pre-trained models, datasets, and data transformation utilities for computer vision and deep learning tasks.
- Albumentations [72]: A library for image augmentation techniques that we employed to enhance the diversity of our dataset.
- Torchmetrics [73]: A library for evaluation metrics that we used to assess the performance of our models.

In addition to the aforementioned libraries and software setup, we utilized the following technologies to enhance the performance and efficiency of our experiments:

- NVIDIA CUDA v11.6 [74]: A parallel computing platform and API model that allows us to leverage the power of NVIDIA GPUs for accelerated computation in deep learning tasks.
- CuDNN [75]: A GPU-accelerated library for deep neural networks that provide highly optimized implementations of primitive functions, such as convolutions and pooling, to maximize the efficiency of deep learning models.
- Anaconda [76] (conda 22.11.1): A popular Python distribution that simplifies the management of software packages and environments. We used Anaconda to create and manage our Python environment, ensuring the compatibility and reproducibility of our experiments.

3.3 Hardware Environment

The experimental setup utilized the following hardware environment:

- Processor: 11th Gen Intel(R) Core(TM) i7-11800H @ 2.30GHz, 2.30 GHz
- GPU: NVIDIA GeForce RTX 3070 Laptop GPU

- Installed RAM: 16.0 GB (15.8 GB usable)
- OS: Windows 11 64-bit operating system
- Manufacturer: Razer Inc.

3.4 Deep learning model selection

In order to address the limited data availability pre-trained models were selected for the experiments. These models were fine-tuned specifically for water detection and tested with the collected dataset.

Based on our research of deep learning techniques, we chose not to employ the method in the paper [34] designed for stagnant water detection as it generates a heat map that is not suitable for our semantic segmentation task. Furthermore, we experimented with temporal approaches [31] [32] but did not observe significant progress during the training phase, leading us to the decision of discontinuing their use in our experiment. We also did not include bounding box approachs.

We only employed FCN-8s-FL-5RAU [4] as a more relible and relevant exisitng method. This choice was based on its suitability for our water detection task and its proven performance in semantic segmentation.

The following networks were utilized:

- **DeepLabV3** [65]: Semantic segmentation model with a ResNet101 [52] backbone Pythorch provides pre-trained weights for DeepLabV3. The training of these weights was performed on a portion of the COCO dataset [55], specifically focusing on the 20 categories that are also found in the Pascal VOC [54] dataset. On the COCO validation dataset, it demonstrates a high Mean Intersection over Union (M-IoU) of 0.67. The model has 60 996 202 parameters. [77]

- **Mask R-CNN** [61]: Instance Segmentation model with Resnet50 FPN backbone:

  Pythorch also provides pre-trained weights for Mask R-CNN. The training of these weights was performed on the COCO dataset with 91 categories including human, bicycle and etc. These weights were generated through an improved training process aimed at enhancing the accuracy of the model.[78] It is reported by Pytorch that it shows 34.6 mask mean Average Precision (mask mAP) on the COCO validation dataset [79]. The model has 44 401 393 parameters.

- **ResNet101** [52]: Instance Segmentation model:

  It is one of the pre-trained semantic segmentation networks provided by PyTorch. It is trained on ImageNet [56] dataset with 1,000 object categories. It shows 93.546% accuracy on the ImageNet test dataset. The model has 44 549 160 parameters. [80]

- **FCN-8s-FL-5RAU** [4]: A semantic segmentation model.

  It is provided by the authors (Han et al. 2018) to detect water hazards in on-road and off-road environments. We have implemented the model with the Pythorch framework and compared its result with other networks. it is reported in the paper [4] that the model shows F-measure 76.91%, Precision 78.03%, Recall 75.81%,

and Accuracy 99.34% on both off-road and on-road Puddle-1000 [4] test datasets. The model has 208 762 902 parameters.

3.5 Model Training

The model training process in this thesis utilized the following steps:

Optimization Algorithm: The decision to use the Adam optimizer [81] and the Stochastic Gradient Descent [82] optimizer was based on their widespread adoption and reputation for achieving fast convergence. The Adam optimizer demonstrated the highest scores on the validation dataset when used with pre-trained networks. However, for Mask R-CNN, the Stochastic Gradient Descent optimizer [82] was chosen as it achieved slightly better results compared to the Adam optimizer specifically for Mask R-CNN.

Loss Function: The Binary cross-entropy with logits loss [83] was chosen as the loss function for pre-trained models. This particular loss function is specifically designed for binary classification problems, which makes it a suitable choice for our scenario where the output represents a single probability value. For the FCN-8s-FL-5RAU model, we kept the focal loss [84] as stated in the referenced paper. Compared to the Binary cross-entropy with logits loss, this loss function demonstrates higher performance on the validation dataset.

Learning Rate Initialization: To dynamically adjust the learning rate during training, we utilized the StepLR scheduler [85]. This scheduler reduces the learning rate by a factor of gamma after a specific number of training steps (3 in our case). By incorporating the StepLR scheduler into our training process, we ensured that the learning rate was dynamically adjusted to optimize model performance.

Performance Metric: The Intersection over Union (IoU) also known as the Jaccard Index [86], Precision, Recall, F1-score, and Accuracy metrics, was employed to evaluate the performance of the semantic segmentation models.

Data Augmentation: To enhance the generalization capabilities of the models, data augmentations such as Horizontal Flip and Random Brightness were applied during the training phase. These augmentations introduce variations in the training data, enabling the models to learn and adapt to different real-world scenarios. Additionally, we have applied resizing and normalization, which can provide advantages such as standardizing data dimensions and scales and improving comparability across samples. Since the water is expected to be on the ground and positioned horizontally, data augmentations such as vertical flipping, rotation, and padding may not have any significance in this context.

Hyperparameters for all the models are below. Image size is chosen 360x640 for all the models.

- FCN-8s-FL-5RAU: initial learning rate: 0.0001, number of workers: 4, batch size: 1 (higher batch size causes memory overflow in the system).

- ResNet101: initial learning rate: 0.00001, number of workers: 4, batch size: 4.
- Mask R-CNN: initial learning rate: 0.0001, number of workers: 4, batch size: 4.
- DeepLabV3: initial learning rate: 0.00001, number of workers: 4, batch size: 4.

We have modified the last layer of the pre-trained DeepLabV3, converting the classifier into a 1-channel model for binary segmentation. For ResNet101, we have made adjustments to the classifier layer, specifically configuring it to produce a single output channel, and trained only the classifier layer. We have implemented the FCN-8s-FL-5RAU model as stated in the paper [4] with the Pythorch framework and trained it from scratch with all layers. For the Mask R-CNN model, we only trained the mask predictor layer.

Throughout the training, the progress of the Accuracy, Recall, Precision, and F1-score was tracked and they are given in Appendix A. Each model was trained for 8 epochs. During the training process, we saved the models after each epoch and selected the version with the highest Jaccard index for testing. Progress of Jaccard Index (M-IoU) of validation dataset after each epoch during training of DeepLabV3, Mask R-CNN, ResNet101, FCN-8s-FL-5RAU was given in Figure 7.

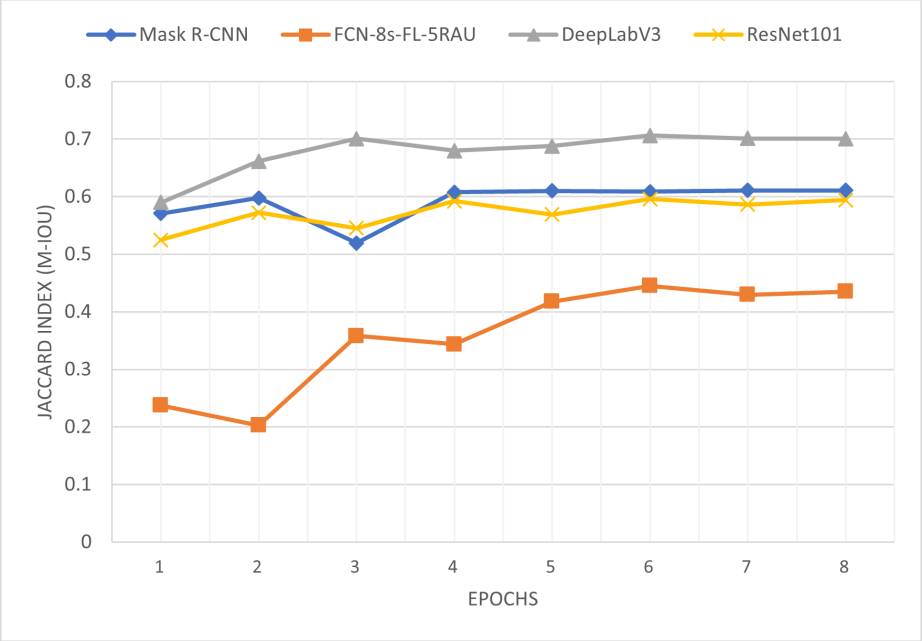


Figure 7. Progress of Jaccard Index (M-IoU) of validation dataset after each epoch during training of DeepLabV3, Mask R-CNN, ResNet101, FCN-8s-FL-5RAU.

Table 1 presents the training time for eight epochs across all models.

Table 1. Training time of the models.

Model	Training time	Average time for one iteration
FCN-8s-FL-5RAU	7 hours	0.3906s
ResNet101	1 hour	0.1461s
Mask R-CNN	3 hours	0.6528s
DeepLabV3	1 hour	0.2028s

4 Results and Analysis

In this section, we have summarised the results from each model on the test dataset.

DeepLabV3: The images showing the results of DeepLabV3 on the test dataset can be found in Figure 8. The analysis reveals that the model faces challenges in identifying puddles. However, it is relatively successful in segmenting water areas in images containing dense vegetation. Additionally, the model demonstrates good performance in detecting sea regions.

The model also has shown some False-positive results in Figure 9.

Figure 8. Superimposed results of DeepLabV3 model on the test dataset.

Figure 9. False segmentation of DeepLabV3 in the water detection.

Mask R-CNN: The result images are shown in Figure 10 for Mask R-CNN on the test dataset. From the results, we can say the model accurately segmented water bodies. The model effectively detected the presence of puddles on the road and identified water within dense vegetation. Additionally, it demonstrated better alignment with the curves of the water area present in the image.

The false positive results are shown in Figure 11. Specifically, It mistakenly classified a car and stones as water during the segmentation process.

Figure 10. Superimposed results of Mask R-CNN model on the test dataset.

Figure 11. False segmentation of Mask R-CNN in the water detection.

ResNet101: The result images of ResNet101 are shown in Figure 12 on the test dataset. Based on the results, it can be observed that the model had success in accurately segmenting puddle and sea areas. However, it faced challenges when dealing with images containing dense vegetation, resulting in scattered or fragmented masks.

In Figure 13, we can see some False positive results by Resnet101. Specifically, it identified the sky as a water area.

Figure 12. Superimposed results of ResNet101 model on the test dataset.

Figure 13. False segmentation of ResNet101 in the water detection.

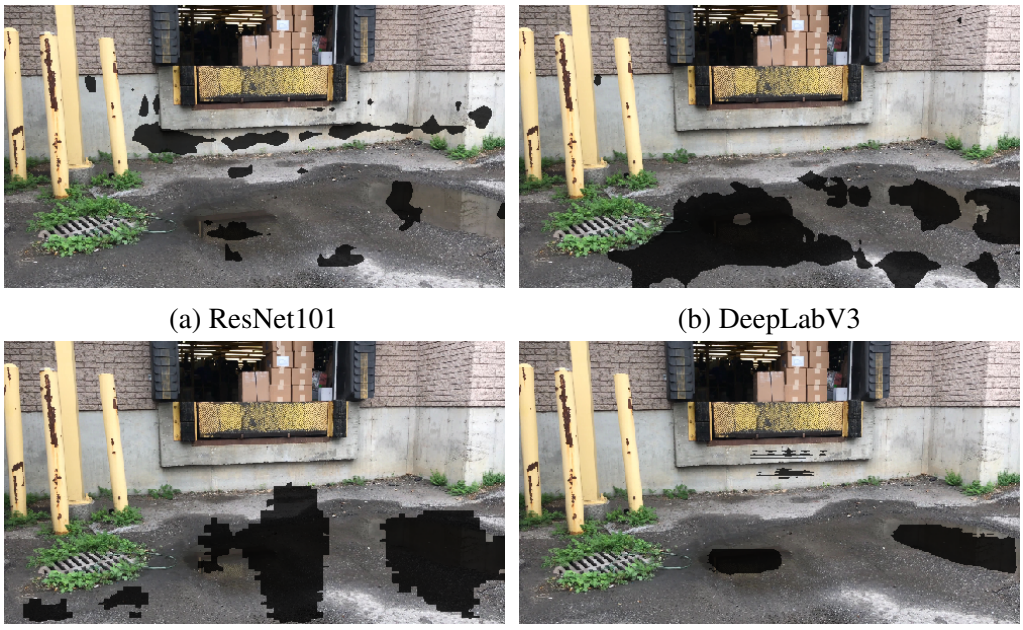
FCN-8s-FL-5RAU: Figure 14 presents the result images obtained from the test dataset. The analysis of these images reveals that the model encounters difficulties when it comes to accurately identifying puddle areas. Additionally, it struggles to segment water areas in scenarios involving dense vegetation. It is important to note that the segmentation mask produced by the model appears to be pixelated and scattered.

In Figure 15, some of the False positive segmentations are shown. The model incorrectly selected the asphalt as a water area.

Figure 14. Superimposed results of FCN-8s-FL-5RAU model on the test dataset.

Figure 15. False segmentation of FCN-8s-FL-5RAU in the water detection.

Due to the limited number of puddle images in the training set, all the models face challenges in accurately segmenting these images. However, among the tested models, Mask R-CNN demonstrates superior performance in accurately segmenting the water area in the puddle images Figure 16.



(a) ResNet101 (b) DeepLabV3

(c) FCN-8s-FL-5RAU (d) MaskRCNN

Figure 16. Superimposed results of all models in the puddle images.

The table provided, labeled as Table 2, showcases the performance metrics of all the models on the test dataset.

- Among the models, Mask R-CNN obtained the highest Jaccard Index of 0.50, suggesting its superior performance in accurately segmenting the objects of interest.

Additionally, it achieved the highest accuracy score of 92.74%, indicating its ability to make accurate predictions in general.

- ResNet101 achieved the highest precision score of 0.70, suggesting a higher proportion of correct positive predictions compared to other models.
- DeepLabV3 achieved the highest F1 score of 0.55, suggesting a good balance between precision and recall.
- The time taken for a single prediction is also reported in the table. Among the models, Mask R-CNN had the longest prediction time of 0.0744 seconds, while FCN-8s-FL-5RAU had the shortest prediction time of 0.0188 seconds.

Table 2. Metrics scores of the models on the test dataset

model	Jaccard index	Recall	Precision	F1 score	Accuracy	Time for a single prediction
FCN-8s-FL-5RAU	0.33	0.35	0.65	0.41	89.98%	**0.0188s**
ResNet101	0.41	0.44	**0.70**	0.51	91.79%	0.0369s
Mask R-CNN	**0.50**	**0.55**	0.37	0.43	**92.74%**	0.0744s
DeepLabV3	0.46	0.53	0.66	**0.55**	92.55%	0.0220s

4.1 Discussion

Based on the findings, it is evident that pre-trained models yield superior outcomes. These models demonstrate stronger generalization ability, as the test data comprises previously unseen data. Additionally, FCN-8s-FL-5RAU which is trained from scratch tend to struggle in accurately delineating water bodies. Instead, it produces scattered outcomes.

Furthermore, the training time for the pre-trained models is significantly reduced since we only trained the last layer of the networks. This allows for efficient fine-tuning and faster monitoring of metric improvements.

It should be noted that the training dataset may not be fully representative of the validation and test datasets. This is primarily due to a limited amount of data available from the UGV to populate the dataset. As a result, a significant portion of our custom data was allocated to the validation and test sets in order to evaluate the model's performance in our specific environment.

The models face challenges when it comes to detecting unseen water bodies. Specifically, they struggle with identifying small puddles in the images. However, they perform relatively better when it comes to detecting seashores. It is important to note that different types of water bodies, such as muddy puddles, stagnant water, and larger bodies like

lakes or seas, pose varying levels of difficulty for the models. Additionally, achieving accurate results becomes more challenging when the water bodies are surrounded by vegetation.

4.2 Conclusion

This thesis has explored the use of deep learning techniques to develop a water detection system for UGVs in off-road environments. The proposed approach combines convolutional neural networks with transfer learning techniques to achieve more accurate water detection capabilities. The thesis includes a literature review of existing water detection methods, an evaluation of different CNN architectures with transfer learning and training strategies, and a validation of the proposed approach using real-world data.

Based on our research and findings, it can be concluded that sensor technologies such as lidar and sensor fusing offer a more robust water detection system. These sensors exhibit the capability to detect water in adverse weather conditions and are effective even during nighttime. However, we need to consider that implementation of lidar and sensor fusing systems can be expensive. In our specific case, utilizing the existing camera mounted on top of the UGV which is also used for other object detection tasks are practical and cost-effective for water detection.

The results from the experiments demonstrate that the proposed approach outperforms existing deep learning methods in terms of accuracy and robustness, making it a reliable method for detecting water bodies in off-road environments using deep learning techniques. The work has potential applications in industries such as agriculture and mining where UGVs require water detection capabilities to navigate challenging environments.

In conclusion, this thesis has contributed to advancing the field of autonomous navigation for UGVs by providing a practical solution to an important perception problem. The proposed approach has demonstrated significant improvements over existing methods, highlighting the potential benefits of combining deep learning techniques with transfer learning to improve the performance of CNN models on water detection tasks.

5 Future Work

In order to overcome the challenge of limited data availability, one possible direction for future research is to explore the use of synthetic data to train deep learning models. This approach has the potential to improve the models' ability to detect water sources in off-road environments that have not been seen before.

To address the issue of false positive detections in pre-trained models, it would be beneficial to incorporate water detection with other object detection tasks for off-road scenarios. By identifying other objects in the environment, the model could better distinguish between water and non-water regions. Currently, the proposed solution treats all other objects as background, which may lead to false detections in complex environments. Therefore, incorporating additional object detection capabilities could improve the models' performance and reduce false positives.

Another important aspect to consider in future work is the impact of adverse weather conditions on water detection performance. The proposed approach may not perform well in situations with heavy rain, fog, or other weather conditions that affect visibility.

A Appendix A

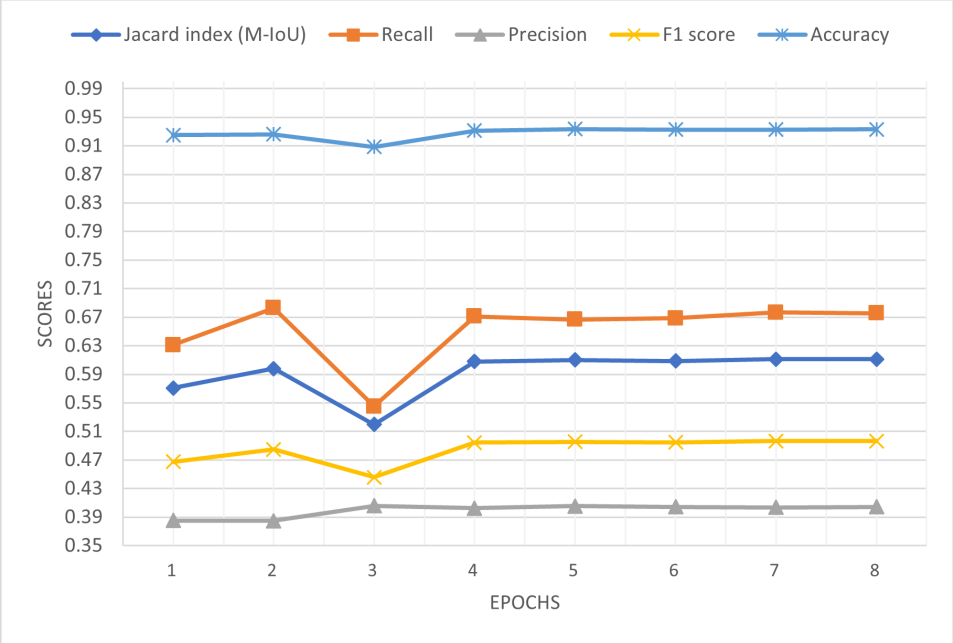


Figure 17. Progress of Jaccard Index (M-IoU), Rrcall, Precision, F1-score, and accuracy of validation dataset after each epoch during training of Mask R-CNN.

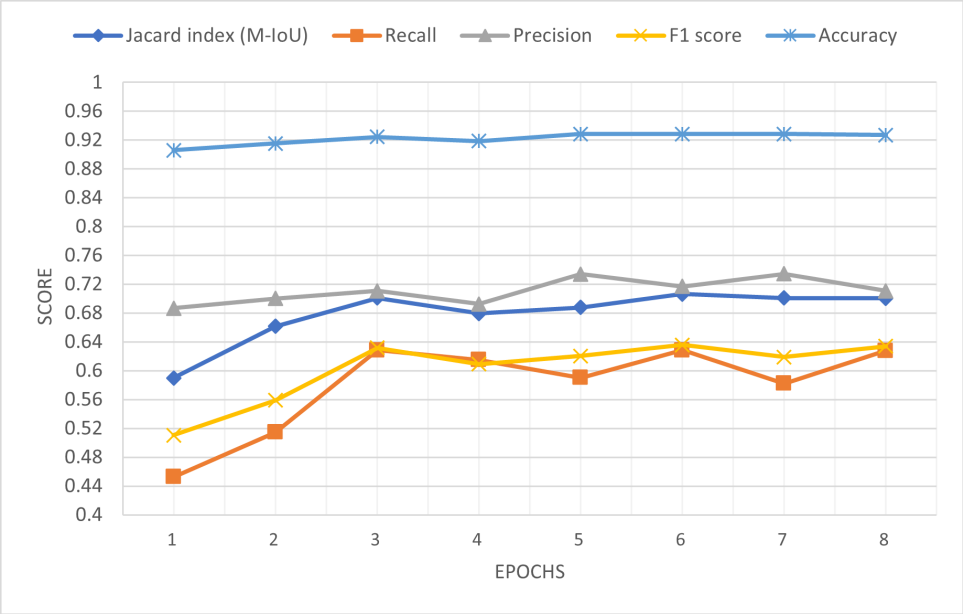


Figure 18. Progress of Jaccard Index (M-IoU), Rrcall, Precision, F1-score, and accuracy of validation dataset after each epoch during training of DeepLabV3.

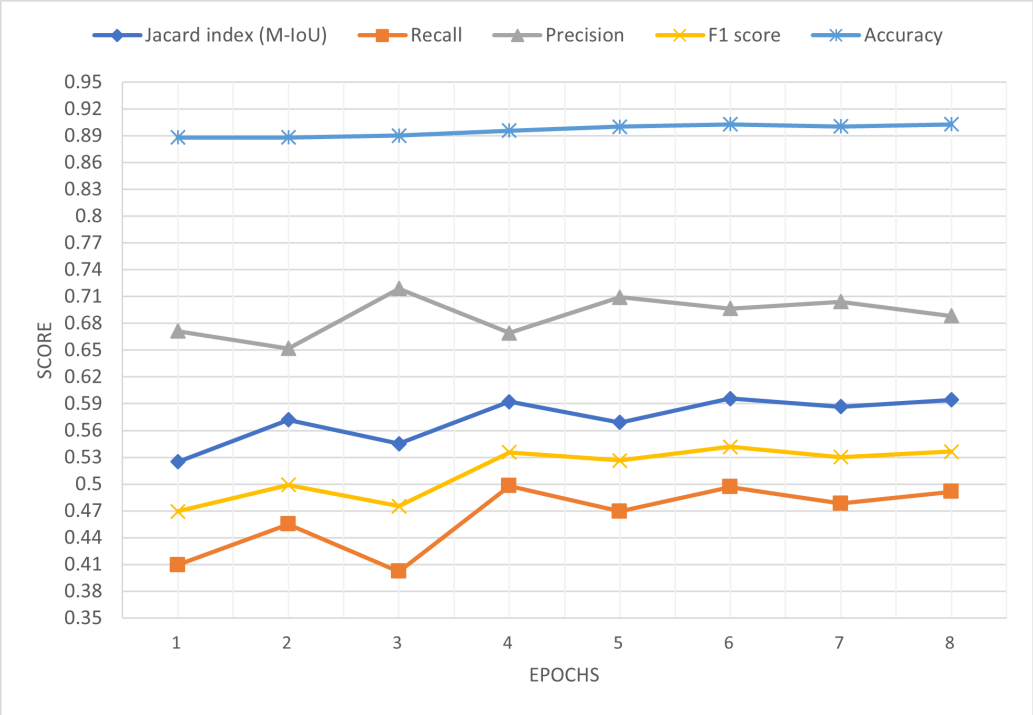


Figure 19. Progress of Jaccard Index (M-IoU), Rrcall, Precision, F1-score, and accuracy of validation dataset after each epoch during training of ResNet101.

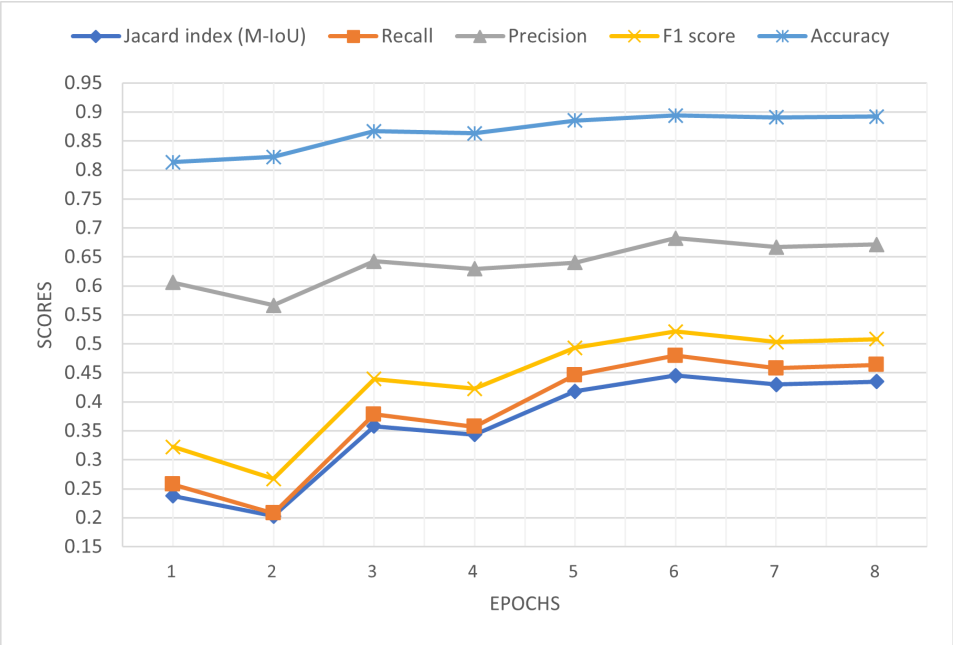


Figure 20. Progress of Jaccard Index (M-IoU), Rrcall, Precision, F1-score, and accuracy of validation dataset after each epoch during training of FCN-8s-FL-5RAU.

B Appendix B

The source code used in this work is in water_detection.zip file.

References

[1] Scikit. https://scikitlearn.org/stable/auto_examples/model_selection/plot_precision_recall.html, 2023. Accessed: May 14, 2023.

[2] Russ Tedrake. Robotic manipulation. Course Notes for MIT 6.4210 http://manipulation.mit.edu, 2022. Accessed: May 10, 2023.

[3] Seyed Mohammad Hassan Erfani, Zhenyao Wu, Xinyi Wu, Song Wang, and Erfan Goharian. Atlantis: A benchmark for semantic segmentation of waterbody images. *Environmental Modelling & Software*, page 105333, 2022.

[4] Xiaofeng Han, Chuong Nguyen, Shaodi You, and Jianfeng Lu. *Single Image Water Hazard Detection Using FCN with Reflection Attention Units: 15th European Conference, Munich, Germany, September 8–14, 2018, Proceedings, Part VI*, pages 105–121. 09 2018.

[5] Ash Rossiter. Bots on the ground: an impending ugv revolution in military affairs? *Small Wars & Insurgencies*, 31(4):851–873, 2020.

[6] D Tilbury and A. Ulsoy. Reliable operations of unmanned ground vehicles: Research at the ground robotics reliability center. 01 2010.

[7] Abhijit Gadekar, Sakshi Fulsundar, Prathamesh Deshmukh, Jaideep Aher, Kaajal Kataria, Vibha Patel, and Dr. Shivprakash Barve. Rakshak: A modular unmanned ground vehicle for surveillance and logistics operations. *Cognitive Robotics*, 3, 03 2023.

[8] Sijo Thomas and Aruna Devi. Design and implementation of unmanned ground vehicle (ugv) for surveillance and bomb detection using haptic arm technology. In *2017 International Conference on Innovations in Green Energy and Healthcare Technologies (IGEHT)*, pages 1–5, 2017.

[9] Liu Xin and Dai Bin. The latest status and development trends of military unmanned ground vehicles. In *2013 Chinese Automation Congress*, pages 533–537, 2013.

[10] Milrem Robotics. The themis ugv. https://milremrobotics.com/defence/, 2023. Accessed: May 5, 2023.

[11] Mustafa Yagimli and H. Selcuk Varol. Mine detecting gps-based unmanned ground vehicle. In *2009 4th International Conference on Recent Advances in Space Technologies*, pages 303–306, 2009.

[12] Simon Monckton. Robotics and military operations. Technical report, Strategic Studies Institute, US Army War College, 2018.

[13] Muhammad Sanaullah, Md Akhtaruzzaman, and Md Altab Hossain. Land-robot technologies: The integration of cognitive systems in military and defense. *NDC E-JOURNAL*, 2(1):123–156, Jan. 2022.

[14] Hirotaka Tahara, Ibuki Ikegami, Kenichiro Takakura, Tatsuya Kato, and Masanobu Nagata. Puddle detection for avoidance path planning of wheeled mobile robot using laser reflection intensity. In *IECON 2019 - 45th Annual Conference of the IEEE Industrial Electronics Society*, volume 1, pages 699–704, 2019.

[15] Hai Shao, Zhen Zhang, Ke Li, Jian Wang, Tao Xu, Shuai Hou, and Liang Zhang. Water hazard detection based on 3d lidar. *Applied Mechanics and Materials*, 668-669:1174–1177, 10 2014.

[16] Kailun Yang, Kaiwei Wang, Ruiqi Cheng, Weijian Hu, Xiao Huang, and Jian Bai. Detecting traversable area and water hazards for the visually impaired with a prgb-d sensor. *Sensors*, 17(8), 2017.

[17] Kailun Yang, Luis M. Bergasa, Eduardo Romera, Juan Wang, Kaiwei Wang, and Elena López. Perception framework of water hazards beyond traversability for real-world navigation assistance systems. In *2018 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 186–191, 2018.

[18] Chuong V. Nguyen, Michael Milford, and Robert Mahony. 3d tracking of water hazards with polarized stereo cameras. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5251–5257, 2017.

[19] Jisu Kim, Jeonghyun Baek, Hyukdoo Choi, and Euntai Kim. Wet area and puddle detection for advanced driver assistance systems (adas) using a stereo camera. *International Journal of Control, Automation and Systems*, 14:263–271, 02 2016.

[20] Anass Mançour Billah, Abdenbi Abenaou, El Hassan Ait Laasri, and Dris Agliz. Water recognition and segmentation in the environment using a spatiotemporal approach. *Pattern Recognition and Image Analysis*, 31:295–312, 04 2021.

[21] Arturo L. Rankin, Larry H. Matthies, and Andres Huertas. Daytime water detection by fusing multiple cues for autonomous off-road navigation. 2006.

[22] Pedro Santana, Ricardo Mendonça, and José Barata. Water detection with segmentation guided dynamic texture recognition. In *2012 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 1836–1841, 2012.

[23] Pascal Mettes, Robby Tan, and Remco Veltkamp. Water detection through spatio-temporal invariant descriptors. *Computer Vision and Image Understanding*, 154, 11 2015.

[24] Haiyan Shao, Zhenhai Zhang, and Kejie Li. Research on water hazard detection based on line structured light sensor for long-distance all day. In *2015 IEEE International Conference on Mechatronics and Automation (ICMA)*, pages 1785–1789, 2015.

[25] Ioannis Katramados, Steve Crumpler, and Toby P. Breckon. Real-time traversable surface detection by colour space fusion and temporal analysis. In Mario Fritz, Bernt Schiele, and Justus H. Piater, editors, *Computer Vision Systems*, pages 265–274, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[26] Thiago Rateke, Karla Justen, and Aldo Von Wangenheim. Road surface classification with images captured from low-cost camera - road traversing knowledge (rtk) dataset. 26, 12 2019.

[27] Thiago Rateke and Aldo Von Wangenheim. Road surface detection and differentiation considering surface damages. 06 2020.

[28] Ravpreet Kaur and Sarbjeet Singh. A comprehensive review of object detection with deep learning. *Digital Signal Processing*, 132:103812, 2022.

[29] Jun Deng, Xiaojing Xuan, Weifeng Wang, Zhao Li, Hanwen Yao, and Zhiqiang Wang. A review of research on object detection based on deep learning. *Journal of Physics: Conference Series*, 1684:012028, 11 2020.

[30] Li Liu, Wanli Ouyang, Xiaogang Wang, Paul Fieguth, Jie Chen, Xinwang Liu, and Matti Pietikäinen. Deep learning for generic object detection: A survey. *International Journal of Computer Vision*, 128, 02 2020.

[31] Juntao Li and Chuong Nguyen. Realtime water-hazard detection and visualisation for autonomous navigation and advanced driving assistance. In *2019 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, pages 287–288, 2019.

[32] Juntao Li, Chuong Nguyen, and Shaodi You. Temporal 3d fully connected network for water-hazard detection. In *2019 Digital Image Computing: Techniques and Applications (DICTA)*, pages 1–5, 2019.

[33] Mehmet Bilge Han Taş, Muhammed Coşkun Irmak, Sedat Turan, and Abdulsamet Haşıloğlu. Real-time puddle detection using convolutional neural networks with unmanned aerial vehicles. In *2021 6th International Conference on Computer Science and Engineering (UBMK)*, pages 598–602, 2021.

[34] Zihao Zhao and Haigang Zhang. A localization method for stagnant water in city road traffic image. *Multimedia Tools and Applications*, pages 1–14, 2022.

[35] Li Deng and Dong Yu. *Deep Learning: Methods and Applications*. 2014.

[36] Yann LeCun, Y. Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521:436–44, 05 2015.

[37] Dan Ciresan, Alessandro Giusti, Luca Gambardella, and Jürgen Schmidhuber. Deep neural networks segment neuronal membranes in electron microscopy images. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.

[38] Dan Ciresan, Ueli Meier, Jonathan Masci, and Jurgen Schmidhuber. A committee of neural networks for traffic sign classification. *Proceedings of the International Joint Conference on Neural Networks*, pages 1918–1921, 07 2011.

[39] Dan Claudiu Cireşan, Ueli Meier, Luca Maria Gambardella, and Jürgen Schmidhuber. Deep, Big, Simple Neural Nets for Handwritten Digit Recognition. *Neural Computation*, 22(12):3207–3220, 12 2010.

[40] li Deng, Jinyu Li, Jui-Ting Huang, Kaisheng Yao, Dong Yu, Frank Seide, Michael Seltzer, Geoff Zweig, Xiaodong He, Jason Williams, Yifan Gong, and Alex Acero. Recent advances in deep learning for speech research at microsoft. pages 8604–8608, 10 2013.

[41] Geoffrey Hinton, Li Deng, Dong Yu, George E. Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N. Sainath, and Brian Kingsbury. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97, 2012.

[42] Ronan Collobert, Jason Weston, Leon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493–2537, 02 2011.

[43] Richard Socher, Jeffrey Pennington, Eric Huang, Andrew Ng, and Christopher Manning. Semi-supervised recursive autoencoders for predicting sentiment distributions. pages 151–161, 01 2011.

[44] Raia Hadsell, Pierre Sermanet, Marco Scoffier, Ayse Erkan, Koray Kavackuoglu, Urs Muller, and Yann Lecun. Learning long-range vision for autonomous off-road driving. *Journal of Field Robotics*, 26:120–144, 01 2009.

[45] Clément Farabet, Camille Couprie, Laurent Najman, and Yann LeCun. Scene parsing with multiscale feature learning, purity trees, and optimal covers. *CoRR*, abs/1202.2160, 2012.

[46] IBM. What is a neural network? `https://www.ibm.com/topics/neural-networks`, 2023. Accessed: May 14, 2023.

[47] IBM. What are convolutional neural networks? `https://www.ibm.com/topics/convolutional-neural-networks`, 2023. Accessed: May 8, 2023.

[48] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 886–893 vol. 1, 2005.

[49] Ravpreet Kaur and Sarbjeet Singh. A comprehensive review of object detection with deep learning. *Digital Signal Processing*, 132:103812, 2022.

[50] Xiao Youzi, Zhiqiang Tian, Jiachen Yu, Yinshu Zhang, Shuai Liu, Shaoyi Du, and Xuguang Lan. A review of object detection based on deep learning. *Multimedia Tools and Applications*, 79, 09 2020.

[51] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv 1409.1556*, 09 2014.

[52] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. pages 770–778, 06 2016.

[53] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

[54] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88:303–338, 2010.

[55] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 740–755, Cham, 2014. Springer International Publishing.

[56] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[57] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361, 2012.

[58] Jianxiong Xiao, Krista Ehinger, James Hays, Antonio Torralba, and Aude Oliva. Sun database: Exploring a large collection of scene categories. *International Journal of Computer Vision*, 119, 08 2014.

[59] Devvi Sarwinda, Radifa Hilya Paradisa, Alhadi Bustamam, and Pinkie Anggia. Deep learning in image classification using residual network (resnet) variants for detection of colorectal cancer. *Procedia Computer Science*, 179:423–431, 2021.

[60] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[61] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. Mask r-cnn. pages 2980–2988, 10 2017.

[62] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. pages 1–10, 01 2016.

[63] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP, 06 2016.

[64] Fahad Lateef and Yassine Ruichek. Survey on semantic segmentation using deep learning techniques. *Neurocomputing*, 338:321–348, 2019.

[65] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv:1706.05587*, 2017.

[66] Tensorflow. Deeplab. `https://github.com/tensorflow/models/tree/master/research/deeplab`, 2023. Accessed: May 8, 2023.

[67] Manpreet Singh Minhas. Transfer learning for segmentation using deeplabv3 in pytorch. `https://towardsdatascience.com/transfer-learning-for-segmentation-using-deeplabv3-in-pytorch-f770863d6a42`, 2023. Accessed: May 17, 2023.

[68] Manpreet Singh Minhas and John S. Zelek. Anomaly detection in images. *CoRR*, abs/1905.13147, 2019.

[69] Guido Van Rossum and Fred L. Drake. *Python 3 Reference Manual*. CreateSpace, Scotts Valley, CA, 2009.

[70] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[71] Daniel Falbel. *torchvision: Models, Datasets and Transformations for Images*, 2023. https://torchvision.mlverse.org, https://github.com/mlverse/torchvision.

[72] A. Buslaev, A. Parinov, E. Khvedchenya, V. I. Iglovikov, and A. A. Kalinin. Albumentations: fast and flexible image augmentations. *ArXiv e-prints*, 2018.

[73] Nicki Skafte Detlefsen, Jiri Borovec, Justus Schock, Ananya Harsh Jha, Teddy Koker, Luca Di Liello, Daniel Stancl, Changsheng Quan, Maxim Grechkin, and William Falcon. Torchmetrics - measuring reproducibility in pytorch. *Journal of Open Source Software*, 7(70):4101, 2022.

[74] NVIDIA, Péter Vingelmann, and Frank H.P. Fitzek. Cuda, release: 10.2.89. `https://developer.nvidia.com/cuda-toolkit`, 2020. Accessed: May 14, 2023.

[75] NVIDIA Developer. Nvidia cudnn. `https://developer.nvidia.com/cudnn`, Feb 2023. Accessed: May 14, 2023.

[76] Anaconda Inc. Anaconda software distribution. `https://docs.anaconda.com/`, 2020. Accessed: May 17, 2023.

[77] Torch Contributors. deeplabv3 resnet101 - torchvision main documentation. `https://pytorch.org/vision/master/models/generated/torchvision.models.segmentation.deeplabv3_resnet101.html#torchvision.models.segmentation.DeepLabV3_ResNet101_Weights`, 2023. Accessed: May 17, 2023.

[78] Torch Contributors. Mask r-cnn - torchvision main documentation. `https://pytorch.org/vision/main/models/mask_rcnn.html`, 2023. Accessed: May 17, 2023.

[79] Torch Contributors. Mask r-cnn resnet50 fpn. `https://pytorch.org/vision/main/models/generated/torchvision.models.detection.maskrcnn_resnet50_fpn.html#torchvision.models.detection.maskrcnn_resnet50_fpn`, 2023. Accessed: May 17, 2023.

[80] Torch Contributors. Resnet101. https://pytorch.org/vision/main/models/generated/torchvision.models.resnet101.html, 2023. Accessed: May 17, 2023.

[81] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 12 2014.

[82] Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.

[83] Bcewithlogitsloss - pytorch 2.0 documentation. https://pytorch.org/docs/stable/generated/torch.nn.BCEWithLogitsLoss.html, 2023. Accessed: May 17, 2023.

[84] sigmoid focal loss - torchvision main documentation. https://pytorch.org/vision/main/generated/torchvision.ops.sigmoid_focal_loss.html, 2023. Accessed: May 17, 2023.

[85] PyTorch Contributors. Steplr - pytorch 2.0 documentation. https://pytorch.org/docs/stable/generated/torch.optim.lr_scheduler.StepLR.html, 2023. Accessed: May 17, 2023.

[86] Jaccard index - pytorch-metrics 0.11.4 documentation. https://torchmetrics.readthedocs.io/en/stable/classification/jaccard_index.html, 2023. Accessed: May 17, 2023.

Non-exclusive licence to reproduce the thesis and make the thesis public

I, Fidan Rustambayli

1. grant the University of Tartu a free permit (non-exclusive licence) to:

   reproduce, for the purpose of preservation, including for adding to the DSpace digital archives until the expiry of the term of copyright, my thesis

   Comparison of Water Detection Models for an Off-road Unmanned Ground Vehicle,

   supervised by Joosep Kivastik and Mihkel Pajusalu.

2. I grant the University of Tartu the permit to make the thesis specified in point 1 available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, under the Creative Commons licence CC BY NC ND 4.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work from **25/05/2028** until the expiry of the term of copyright,

3. I am aware that the author retains the rights specified in points 1 and 2.

4. I confirm that granting the non-exclusive licence does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

Fidan Rustambayli
25/05/2023