Automatic Segmentation of Ships in Digital Images

A Deep Learning Approach

A.B. van Ramshorst

Master of Science Thesis

TU Delft Delft University of Technology

Delft Center for Systems and Control

Automatic Segmentation of Ships in Digital Images

A Deep Learning Approach

Master of Science Thesis

For the degree of Master of Science in Systems and Control at Delft University of Technology

A.B. van Ramshorst

August 27, 2018

Faculty of Mechanical, Maritime and Materials Engineering (3mE) · Delft University of Technology

The work in this thesis was supported by TNO. A data set was provided by the Maritime Warfare Centre (MWC). Their cooperation is hereby gratefully acknowledged.

Abstract

Knowledge on adversaries during military missions at sea heavily influences decision making, making identification of unknown vessels an important task. Identification of surrounding vessels based on visual data offers an alternative to AIS information (Automatic Identification System), the current standard in vessel identification, which can be spoofed. One visual approach employs human expertise and manually identifies vessels guided by a ship catalog. In order to minimize or potentially eliminate human error and performance limitations, there is strong interest in developing an automated vessel classification pipeline. One such pipeline is currently being developed at TNO, capable of classifying over 500 separate classes. A crucial part of the classification pipeline is retrieving an accurate contour of a vessel from a digital image.

To address this important challenge, this thesis proposes an advanced deep learning pipeline to automatically segment the vessel image into background (e.g. sky and sea) and the object of interest (a vessel). Deep learning models based on Fully Convolutional Neural Networks (FCNs) have achieved high performance on the task of semantic segmentation. Several networks such as CRF-RNN, PSPNet, DeepLab and Mask R-CNN are employed to determine a baseline performance. We will focus on identifying the cause of poor or failing segmentations and aim to construct a robust network capable of handling these challenges. By sampling disturbances, caused by ship distance and camera noise, augmented data sets are built to tune networks to input from on-site images. Additionally, experiments are done to evaluate the influence of different levels of disturbances.

Previous approaches implementing the CRF-RNN network achieved top 1 and top 5 classification accuracies of 31.1% and 44.0% respectively. Employing the DeepLab network, trained to convergence on artificial noise augmented data, we report top 1 and top 5 accuracy of 68.9% and 88.8% respectively. Additionally, implementing an ensemble of classifiers, performance is increased to 73.0% and 91.7% for top 1 and top 5 accuracy respectively. This best result is comparable to the classification results with human annotated ship silhouettes. The human performance accuracy is 73.4% on top 1, and 91.3% on top 5 classification performance. Finally, we show that training on a collection of different levels of image disturbances results in a network that is robust against increasing disturbance in images, while retaining performance on clean images.

Table of Contents

List of Figures

List of Tables

Preface

A project on machine learning algorithms for a computer vision application may seem far removed from the usual projects carried out by master students at the DCSC. Luckily, that did not stop me from pursuing a project which focused heavily on deep learning approaches. Combined with a newly found interest in the field of computer vision, a graduate intern position at the Intelligent Imaging department at TNO proved to be the perfect fit for me.

I would like to thank dr. Klamer Schutte in particular for supervising my work at TNO. Not only did he support me on tackling issues related to my thesis work, his guidance in managing such a project proved invaluable. Furthermore, I would like thank dr. ing. Raf van de Plas for his contributions as supervisor from the DCSC. In addition, I would like to thank Tom Hemmes, Arthur van Rooijen, and Ward Heij for all the inspiring debates and discussions in our office, which were not always related to the topic of Artificial Intelligence.

Chapter 1

Introduction

At TNO, the System for Automatic Classification of EO/IR/ISAR imagery (SACEI) project is developed to assist operators with ship classification. The project is carried out at the request of the Maritime Warfare Centre (MWC). The MWC is an organization within the Dutch Ministry of Defence, which publishes guidebooks, newsletters and operational concepts. The SACEI system is intended to be installed and operated aboard various vessels, for use in operations at sea. In warfare situations, reliable and fast identification of vessels at a large distance is of great importance. The SACEI platform is designed to speed up that process and increase its reliability. While increasing performance in those fields, the SACEI platform reduces the level of operator-intensity on the classification task. This is done by automatically matching an input image with entries in a database and subsequently returning the best matches. However, certain steps in this process require human interaction to achieve satisfactory performance during classification. This thesis is focused on the process phase where a ship silhouette is extracted, for use as a feature in the classification step. More accurately, we focus on automating this step using image segmentation.

The type of input images considered in this thesis are in the Electro-Optical (EO) domain. More specifically, Red Green Blue (RGB) digital images. A ship classification system is in place at TNO within the SACEI project. The mechanics and constraints of this classification system are outside the scope of this project. For this reason we aim to construct an input for the classifier that maximizes its performance.

SACEI

The ship classification algorithm implemented as part of the SACEI project consists of multiple steps. The input is a digital image. In this thesis we focus on RGB images. From this image a descriptive feature for subsequent classification is extracted. The feature used is the silhouette of the ship in the digital image. Subsequently, this silhouette is compared to entries in a database. This database consists of projections of a ship model onto an image. Each ship class is viewed from different angles, resulting in a 2D silhouette for each ship class at every viewing angle. The angle intervals are constrained to exclude top down views

and other ranges that are not considered useful. Provided with a ship silhouette image and a database, the algorithm returns a ranking of ship classes. These are ranked in order of similarity, providing the most likely matches at the top of the ranking. In Figure 1-1 the classification pipeline is shown, including visualizations of every step.

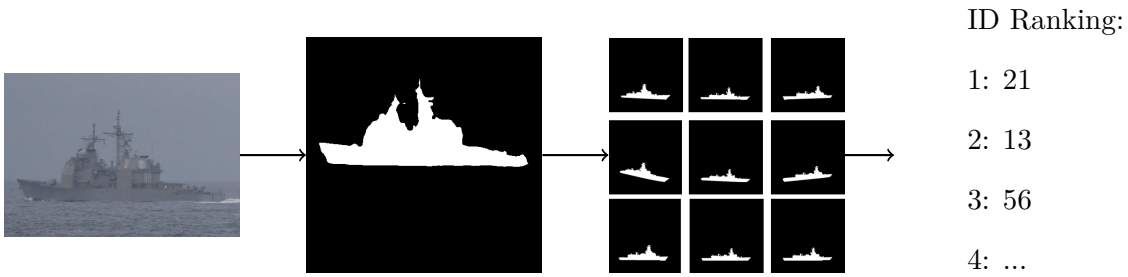


Figure 1-1: The SACEI classification pipeline. From left to right: an input image is presented, a segmentation amp is extracted (showing the ship silhouette), a comparison is made with entries in a database of ship silhouettes, the resulting list of ship IDs in the database is returned (ranked by similarity with the input silhouette).

Automatic Segmentation

The part of the SACEI algorithm described that this thesis will aim to improve, is the extraction of the silhouette feature from a digital image. Currently a deep learning based auto segmentation tool is implemented in the SACEI project. However, the performance of this particular network is not satisfactory. Operators indicate that certain images fail to generate a segmentation, even though visually an image is of acceptable quality. Based on this observation, the detection robustness of deep learning segmentation networks should be explored. A second operator complaint is the omission of ship parts from the segmentation, or inclusion of background patches. To solve this problem, the segmentation accuracy should be improved.

1-1 Problem Statement

Based on operator feedback, we can derive three issues that are of interest in this thesis. The following factors are a source of poor overall performance in the automatic segmentation step:

- **Image Distortion** Due to distance between the optical sensor and the vessel that is observed, several natural phenomena distort the resulting digital image. Combined with this distortion, image sensors introduce distortions of their own. This introduces a challenge for the segmentation algorithm, that is not always robust to handle these types of image distortion.
- **Scale Differences** Images introduced to the classification system do not always contain a ship in a standard resolution. Some input ships are described in an image region with a width of 400 pixels, others might have no more than 100 pixels. The structure of deep learning segmentation algorithms are not always sensitive on different scale levels, introducing a varying performance on different image sizes.

- **Edge Accuracy** To describe the vessel that has to be classified, a silhouette feature is extracted. The higher the precision of this feature, the more descriptive information it holds. A too coarse segmentation can lead to a ship silhouette that lacks discriminatory structures in its contour, leading to poor performance of the classification step.

1-2 Research Questions

An image segmentation algorithm based on the work of Zheng et al. [1] is used for the task of automatic segmentation in the current SACEI pipeline (Figure 1-1). Based on the work of Krizhevsky, Sutskever, and Hinton [2], which we discuss in this thesis, and the subsequent dominance of deep learning in the field of image classification, a deep learning solution to the segmentation problem is proposed. However, not every network is applicable to the same type of segmentation. This means that implementing a deep learning segmentation network is not a straightforward task. A background study is done to understand the mechanics, and their link to the resulting output of a segmentation network. Based on this knowledge, internally different networks are selected for a comparative study. Resulting from that study, the most promising network is implemented and we aim to improve the performance on our domain. We take into account the problems defined in Section 1-1 to focus on different aspects of performance issues for segmentation networks. We divide the research into these two parts, answering two questions with their respective sub questions:

1. **What is the best network structure to apply to the segmentation task?**
   (a) What network structures are used for segmentation in literature?
   (b) How do they compare on the segmentation task?
   (c) How do they compare on the classification task?
   (d) What network is suitable for further research?
2. **Can we train a network to be robust to disturbances present during operation and boost performance?**
   (a) Will fine tuning a general segmentation network on the ship image domain improve performance?
   (b) Does training a network on (modeled) noisy data increase performance on degraded images?
   (c) Does combining different noise levels in a data set reduce the trade-off between performance on clean data and noisy data?
   (d) Does combining multiple network outputs increase performance?

1-3 Thesis Outline

This chapter covered information on the current project, the challenges faced in the segmentation step, and an approach on how to tackle these problems. In Chapter 2 a background

study is done on the mechanics of an Artificial Neural Network (ANN), their application on the image domain, and their use in image segmentation. We also highlight different networks that are comprised of different approaches to increase their segmentation performance. In Chapter 3 the tools used in this thesis are explored. These tools consist of the datasets that are available, a noise model to augment the datasets with disturbances, and metrics to assess the performance of the different networks. In Section 4-1, the described networks from Chapter 2 are considered for a comparative study on their effectiveness on the task of image segmentation. More specifically, their performance on a dataset containing images representative of images acquired during operations at sea. In Section 4-2 the impact of fine tuning a network on our image domain data is explored. Subsequently, the impact of training on data that is augmented with noise is explored. In Section 4-3, the impact of different training and noise settings are evaluated on the SACEI classification algorithm. The experiments are finalized in Section 4-4 and Section 4-6, where an effort is made to use combined datasets and combined classification results respectively to increase performance. In Chapter 5, the research questions are answered. A discussion on the results is added, and recommendations for future work and industry are given.

1-4 Constraints for Public Reporting

This thesis is carried out within a setting that requires a certain level of confidentiality. In light of these restrictions, the contents of databases supplied by the Maritime Warfare Centre are not shown. Images similar to the contents of the supplied database are shown instead. In addition, we do not fully elaborate on the mechanics of the SACEI algorithm. This is avoided due to the mechanics being beyond the scope of this project and to protect the intellectual property of TNO.

Chapter 2

Literature

In this chapter we take a look into the inner mechanics of an Artificial Neural Network (ANN). A brief introduction to neural nets is given, followed by an introduction to the concept of a Convolutional Neural Network (CNN). For the CNN the commonly used layers are investigated, and important operations such as training and weight updates are presented. In Section 2-2, the application of a CNN on the segmentation task is highlighted. Several networks and their structures are presented, exploring the differences between available segmentation networks.

2-1 Introduction to Neural Networks

While the first application of a neural network to the task of image recognition was as early as 1989 [3], it was not a popular image segmentation method in subsequent years due to limitations in computational power and training data availability. In 2010 the ImageNet Large Scale Visual Recognition Challenge (LSVRC) [4] was proposed and went on to become one of the most popular benchmarks in image recognition. In 2010 and 2011, approaches that proved to be most accurate were based off of the image pipeline concept (Figure 2-1) [5], both using a variation implementing the SIFT algorithm [6] for features extraction and a SVM classifier [7] for the classification step.

Figure 2-1: The classical image classification pipeline.

One of the biggest breakthroughs, in the area of performance, came in 2012 with the entry of the SuperVision Team [2]. More commonly known as AlexNet, this neural network approach exceeded the performance of all other algorithms and set the standard for subsequent years. This network was able to assign a class to an image, a single output based on a set of pixels as

an input. Through the use of new architectures, the ability to assign per-pixel labels allowed neural networks to surpass other techniques, not only on the task of image classification but also on the task of image segmentation [8].

2-1-1 Architecture

In Figure 2-2 an example structure of a classic ANN is shown. The most commonly used architecture for image recognition tasks is the CNN. There are a few important differences between a dense ANN and a CNN. The first major difference compared to a regular artificial neural network is the dimension of the input. Where the regular ANN is presented with a one-dimensional vector of values, a network that is built for images has to deal with a two-dimensional input. This is extended to three dimensions in the case of color images, where for example each pixel contains information for the red, green and blue values. A great advantage is that the separate input nodes represent pixels, and therefore contain spatial information. There are three main layer types in a network designed for image recognition [3, 9]. These will be highlighted in the following sections.

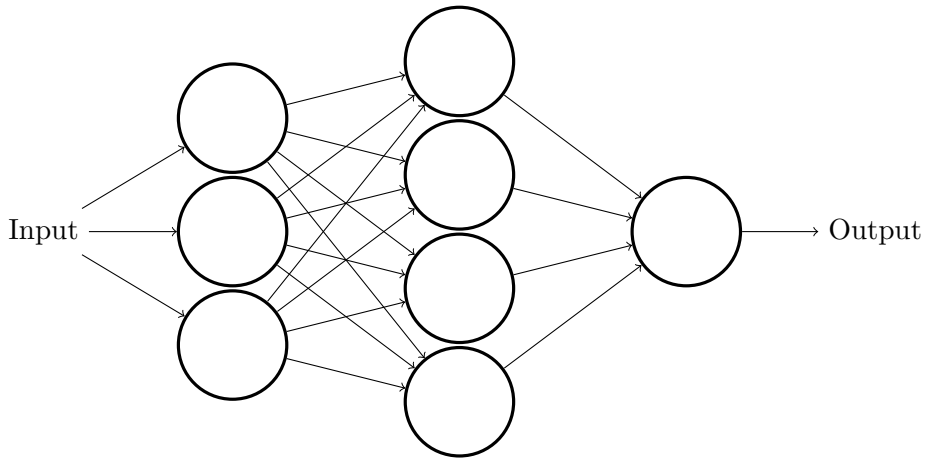


Figure 2-2: Example of an ANN structure, each circle resembling a perceptron as in Figure 2-3.

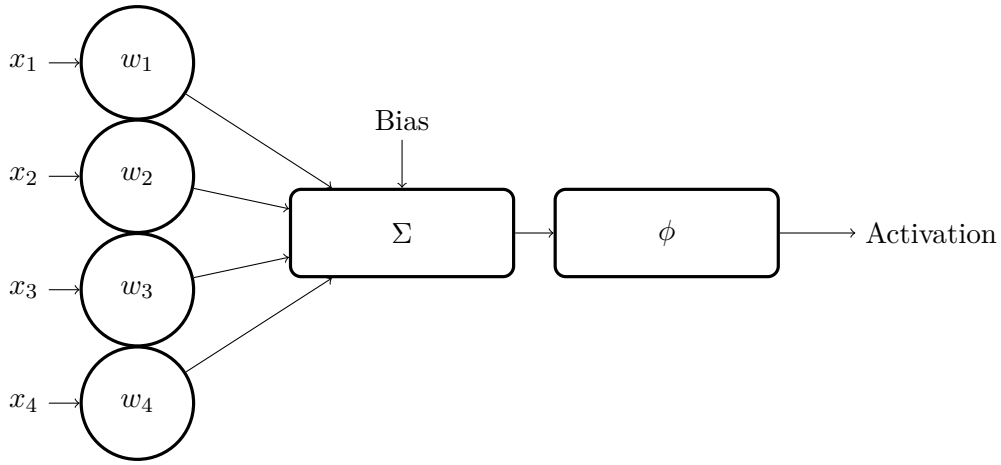


Figure 2-3: Example of a perceptron with four inputs, such as the output node in Figure 2-2. Array x contains the inputs, w represents the weights and ϕ is an activation function.

2-1-2 Convolutional Layers

A neural network that interprets images instead of feature vectors can be built by simply reshaping the image pixel-value vector to an array. This array can be passed to a network similar to Figure 2-2, but with an amount of inputs according to the number of pixels in the image. This solution is sensitive to overfitting due to the large amount of connections between the layers [9]. More importantly, when increasing the image resolution the amount of connections grows, increasing the severity of this problem. A solution is to introduce a convolutional layer. This layer is created by sliding a filter, commonly referred to as kernel, across the input image. The weights, or parameters, of the filter are constant for all positions on the image, greatly reducing the amount of weights in the network. This approach was first introduced by Fukushima [10]. A depiction of a CNN is shown in Figure 2-4. The number of filters per layer is adjustable. The aim of training in a convolutional layer is to find the values in these filters.

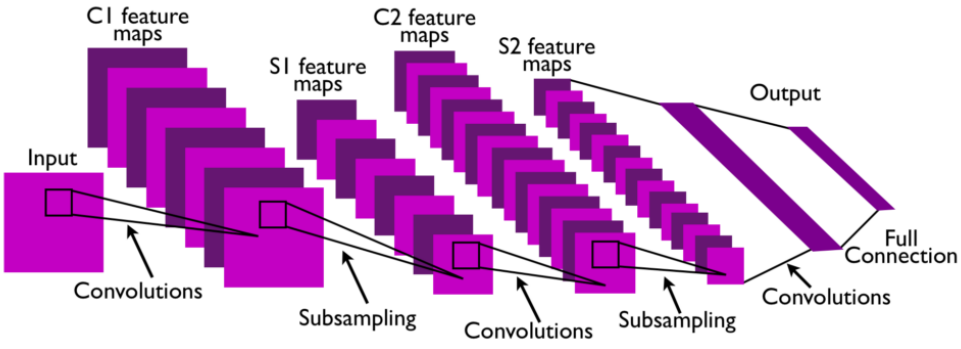


Figure 2-4: Example of the structure of a Convolutional Neural Network [11].

The way that the kernel is moved over the input image and feature maps has a few parameters and properties that can be varied. The first one is the size of the kernel that is moved over

the input. While the depth of the kernel is set by the depth of the input volume, the area can be varied. The size of the step made during each shift, called the stride, does not necessarily have to be 1, but can be increased. To make sure that this step fits neatly over the input, the edges of the input can be augmented with zeros. This is called zero-padding. The number of filters applied over the layer can be varied and is called the depth. The depth allows for multiple feature maps per layer to be generated, enabling the network to detect different features simultaneously.

2-1-3 Pooling Layers

To prevent the increasing growth of the network to lead to overfitting, pooling layers can be implemented to reduce the spatial size of the layers. This also has a positive effect on the computational demands of the network. Pooling is depicted in Figure 2-4 as a subsampling step. The pooling operation is applied to each layer separately. This means that the depth is not affected, only the area of the neuron volume. The most commonly used pooling operation is the max-operator of size 2x2. It effectively breaks the layer up into 2x2 parts and puts the max value to the new smaller layer. Discarding 3 out of 4 values it reduces the size by 75%. Another positive effect is that the pooling layer introduces some invariance to distortions and translations [9]. A drawback is the loss of spatial information. Spatial information is not important to retain when classifying an image. However, it is very important when reconstructing a segmentation map. To counter the negative effects of pooling layers, different structural solutions have been proposed. A summary is shown in Figure 2-5.

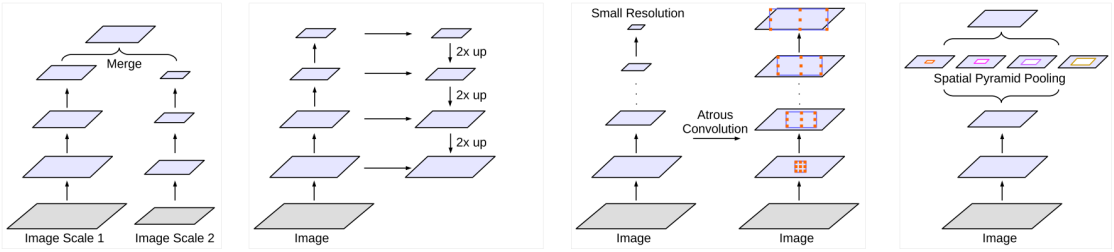


Figure 2-5: Different network structures that are designed to handle the loss of spatial information in the pooling layers. From left to right: Image Pyramid, Encoder-Decoder, Atrous Convolution, Spatial Pyramid Pooling [12].

Four different approaches to the problem of spatial information loss are shown in Figure 2-5. The first one is the image pyramid. This technique is comprised of multiple parallel deep learning pipelines, with their result combined at the end. The input image is rescaled multiple times and presented to the network, allowing the network to detect objects that are small on the image, or cover a large portion. Although this is effective for reducing the effects of differently scaled objects, spatial information is not retained. The second technique shown incorporates interconnections between layers. These interconnections are used to transfer spatial information from early layers to layers that lack this information, due to the pooling operations in between. The third option, implementing atrous convolutions (presented in Section 2-2-3), increases its field-of-view instead of decreasing the field to be evaluated. This way spatial information is retained. The last option is spatial pyramid pooling. It is similar

to an image pyramid, but the parallel evaluation on different scales is due to different pooling operators in the pyramid module, instead of different input images.

2-1-4 Dense Connection Layers

After a series of convolution and pooling layers, a label should be assigned for the classification task. In Figure 2-4 the label is assigned to the image as a whole. In Section 2-2-1 this case will be extended to the segmentation task, with a per-pixel labeled result. During the ImageNet challenge of 2012 the goal was to assign a label to an image as a whole, which was exactly what AlexNet [2] did. Dense connections allow to learn the relations between the detected features and the corresponding labels. Most importantly the mapping from a multidimensional neuron volume to a one-dimensional output vector is realized by dense connection layers.

2-1-5 Training

The action of tuning the weights to a point where they produce the desired output is called training. During training the network is usually presented with an input and a matching output [13]. The inferred output based on the input is compared to the actual, desired output. This can be done with a distance measure designed for the specific use case. Through a loss function (Section 2-1-7) the performance is expressed as a real number. The most common way to improve the network is to compute the gradient at the current point and take steps in the direction of the negative gradient. To apply this principle on each and every layer, backpropagation is used [14]. To allow this method to converge to an optimal point, a sufficient amount of training data must be available. These steps are highlighted in the following sections.

2-1-6 Training Data

The way that the appropriate weights are determined during training is based on supervised learning [13]. To be able to train the high number of parameters in the network, it is essential that the training set is sufficiently large. Specific numbers of training examples needed differ per application and network structure. For example, in Ronneberger, Fischer, and Brox [15] a network is designed to deal with a small example data set for training.

The training data is divided up into multiple sections. One set is meant for training, one set for the validation, and a final set to determine test performance. The test set is used only after the training procedure to ensure there is no bias leading to high performance. The test set is used to tune the network to weights that perform well on the test set itself. During training, a validation set is used to confirm whether the performance is of the same order on a different set of examples. This is important as overfitting on the set of examples is a big drawback that neural networks face. To truly test the performance, a new set that is sufficiently independent from the training data is fed to the network in the test stage. This is the final performance.

2-1-7 Loss Function

The loss function in supervised learning is a mathematical tool to relate the output of the network to the provided ground truth. The result is a real number that signifies a measure of fitness of the network output. A method commonly used in literature is the cross-entropy function [15, 16]. In Equation 2-1 a representation is given as the negative log-likelihood, taking into account the entire list of pixels (which is relevant for image segmentation) [17]. The image is evaluated per-pixel, with a total amount of pixels of N, and i being the pixel number that is considered. The variable y is the array of ground truth pixels, and μ the prediction pixel array.

$$Loss = -\sum_{i=1}^{N} [y_i \log \mu_i + (1 - y_i) \log(1 - \mu_i)] \tag{2-1}$$

While a cost function such as Equation 2-1 is used as an optimization target, it is not necessarily the target of the network. For example, a different performance metric can be used to determine the quality of the output of the network, while still implementing the cross-entropy as a loss function. This is done due to the lack of useful derivatives in most metric functions. In this case the cross-entropy functions as a type of surrogate loss, substituted in for optimization of the network [18].

2-1-8 Backpropagation

During the training procedure in Section 2-1-6, information is gathered about the gradient of the error function with respect to the weights. Based on this information the weights are updated in order to reduce the error [14]. Backpropagation is an efficient method to compute these values. In Equation 2-2 the derivation of the error in the hidden layer is shown for a simple problem. Here δ is the computed error, j is a layer, and k is the layer after j. The weights are represented by w_{kj}, and $h'()$ is the derivative of the activation function. Variable a_j is the sum of weighted inputs for layer j.

$$\delta_j = h'(a_j) \sum_k w_{kj} \delta_k \tag{2-2}$$

The simplified steps below give the derivatives to make a gradient descent step.

1. An example input is presented to the network. Based on Figure 2-3, the outputs of the hidden layers and output layers are calculated.

2. Based on the inferred outputs and the supplied correct outputs in the training set the error is calculated through a loss function.

3. Employing backpropagation (Equation 2-2) the errors in the hidden layers can be computed layer by layer. This starts from the output and works its way back to the input, explaining the name of the method.

4. When all errors are known the derivatives of the error function with respect to the weights can be calculated. A weight update can be done in, for example, the steepest descent direction (depending on the optimization algorithm).

To apply this principle to a convolutional setup, some alterations must be made to account for the dimensionality of the network and the constraint that the filters share weights.

2-1-9 Optimizers

The goal during training is to improve the result of the network, by altering the weight configuration. In the last step of the backpropagation algorithm, the derivatives of the errors with respect to the weights are calculated. If we changed the weights in a way that the error is reduced, we expect that the resulting weight configuration will have a better result. We construct an update rule for the weights, shown in Equation 2-3. In this rule w_k is the new weight, and w_{k-1} the weight of the previous step. The parameter α is the learning rate parameter, which can be tuned and affects the performance. The function $\hat{\nabla} f(w)$ calculates the value that results in a step in the steepest descent, based on the error derivative with respect to the weights.

$$w_k = w_{k-1} - \alpha_{k-1} \hat{\nabla} f(w_{k-1}) \tag{2-3}$$

This particular optimizer rule is called Stochastic Gradient Descent (SGD) [19]. Different improvements were made, in particular with the algorithm called Adam, presented by Kingma and Ba [20], that eliminated the need for an accurately tuned learning rate. However, an accurately tuned SGD optimizer can still outperform Adam optimizers, according to a study by Wilson et al. [21].

2-2 Neural Networks for Image Segmentation

In this section the step from an image classification to an image segmentation network is made. Based on that principal, which is presented in Section 2-2-1, multiple different networks were designed and presented in literature. For our comparative study on the performance of different networks, and how their structure is related to their performance, we explore networks that are diverse in structure. Networks that implement different modules, such as the structures in Figure 2-5, are selected. Besides being significantly different in their basic structure, the networks considered must also be available online. Preferably with a pretrained weight distribution, so we can achieve the results that are reported in literature without having to train the networks. The networks that pass these constraints and are presented in this section are CRF-RNN, DeepLab, PSPNet, and Mask R-CNN.

2-2-1 Fully Convolutional Neural Network (FCN)

The task of image segmentation focuses on the grouping or labeling of each individual pixel. The output layer of the network that is designed for such a task will have to be of the dimensions that correspond to the input image. Instead of subsampling to a coarser feature map the

coarse maps are upsampled back to their original size. This functionality is implemented with a kernel that can be learned during training. In the model presented by Long, Shelhamer, and Darrell [8], a network is used that introduces deconvolutional layers to upsample the feature maps back to the original input size. In Figure 2-6 a single layer is shown upsampling the image.

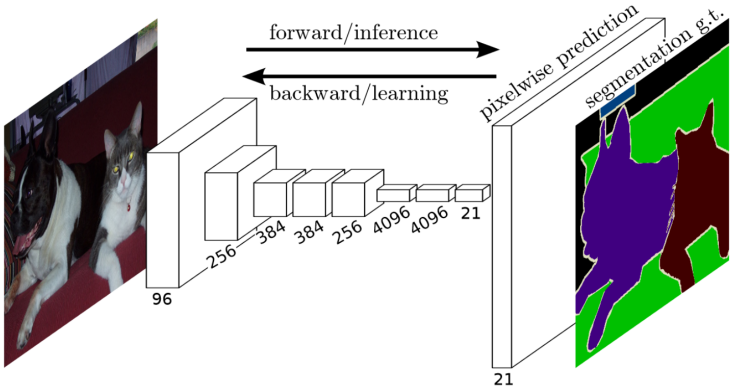


Figure 2-6: Example of the structure of a Fully Convolutional Neural Network, illustrating the input and resulting output [8]. The numbers signify the amount of filters in a layer. In the last layers the 21 filters correspond with the amount of classes. In the output layer each filter represents a class probability heatmap.

When implementing bilinear interpolation in the deconvolutional layers the resulting segmentation is not exact. This means that the edges and boundaries of objects in images are not sharp and the result is a smudgy representation of the image. Compared to the input image, the boundaries are not refined, but the network is able to roughly describe the areas where certain label classes are present. Deeper models with more pooling layers are better at classification, but due to a trade-off with spatial precision, the pixel-level classification decreases [22]. To deal with the inaccurate edges, multiple solutions have been explored to refine them. They are summarized in Figure 2-5.

In Figure 2-6, the information from the convolutional layers is upsampled in one step. As mentioned this results in a poor segmentation with respect to edges and boundaries in an image. To account for the loss of spatial information during pooling connections are made between a convolutional layer and its corresponding deconvolutional layer. This results in a somewhat symmetric network structure. U-net [15] is an example that applies the information from the encoder to the layers in the decoder. The values from parts of the encoder are copied, cropped to size and added to the layers in the decoder. A second approach is presented in the network called SegNet [16]. Instead of passing along the values, this network passes along the indices of the results from the max-pooling layer. This information is used during upsampling to better reconstruct spatial information.

2-2-2 CRF-RNN

The current network being employed for segmentation at TNO is presented in Zheng et al. [1]. The structure of the network model is depicted in Figure 2-7. The network is constructed in two parts. The first part is a front-end that consists of a FCN, as described in Section 2-2-1.

In this front-end the structure is used together with deconvolutional layers to reconstruct probability maps, as presented by Simonyan and Zisserman [23]. Probability maps are per-pixel class probabilities, with a separate map for each class. This results in an output structure with the shape $W \cdot H \cdot N$, where W is the input image width, H is the input image height, and N the number of classes. In a regular FCN the highest class probability would determine the assigned class of the pixel.

Conditional Random Field (CRF)

The second part in the CRF-RNN network, depicted in Figure 2-7, is an extra step implemented to refine the probability maps. This is done by applying a CRF. A CRF in the field of image segmentation is an implementation that aims to model the probability that a pixel is of a certain class [24]. To construct that probability, two terms are evaluated. These terms are the unary and pair-wise potentials, as shown in Equation 2-4. The first term ξ_u are the unary potentials. These potentials are calculated in Krähenbühl and Koltun [25] using the location and color, to determine the probability that a pixel belongs to a certain class. In the case of CRF-RNN, an FCN is used to estimate these class probabilities. The second term, ξ_p, represents the pair-wise potentials. By comparing a pixel i with all other pixels in the image, denoted as j, a more accurate estimation can be made regarding the per-pixel class probability.

$$E(x) = \sum_i \xi_u(x_i) + \sum_{i<j} \xi_p(x_i, x_j) \tag{2-4}$$

The relation between each and every pixel is taken into consideration. With N representing the number of pixels, this results in N^2 relations, which is infeasible to calculate in a straightforward fashion. The approach from Krähenbühl and Koltun [26] approximates these relations, reducing the computation time. The standard computation time in a densely connected CRF is related to the number of pixels, defined as $O(N^2)$. Here, $O()$ is the computation time function and N is the number of pixels. The approximation of the dense CRF scales linearly with time, defined as $O(N)$.

The CRF-RNN layer contains three type of weights classes. In regular CRFs these values are fixed, but in Zheng et al. [1] the assumption is made that learning these weights explicitly would improve performance. The first two sets of trainable weights govern the influence of the spatial kernel and the bilateral kernel. The reason is that certain classes, for example a TV, depend less on a bilateral kernel due to the presence of many different colors in the object. When a class is more likely to have a uniform color, such as airplanes, the weight for the bilateral kernel can be higher. The same principal applies for the spatial kernel. If a class tends to be continuous in shape, the weight for the spatial kernel will be higher. The third weight set governs the compatibility between classes. The compatibility matrix is depicted in Figure 2-8. It shows the likelihood that two similar pixels, in either location or appearance, are assigned a certain class combination. The lower the value, the more likely that two similar pixels are assigned that class pairing. For example, two similar pixels with the same class are assigned a low penalty. Two similar pixels with classes that often appear close, such as a chair and a table, have a low penalty as well. High penalties are given to classes that are

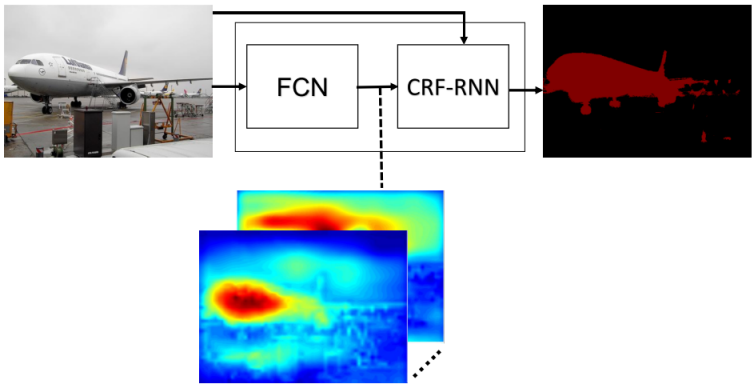


Figure 2-7: Model of the global structure of a FCN combined with an edge/boundary refining step. The refining step is presented with both the original input and the output of the FCN. The output of the FCN consists of a probability map for each class. The output is a per-pixel maximum value taken over all class probability maps. The result is a finer segmentation [1].

unlikely to share the same features, spatial or in appearance. Performance gain over standard weights is reported in Zheng et al. [1], due to the class specific learned weights.

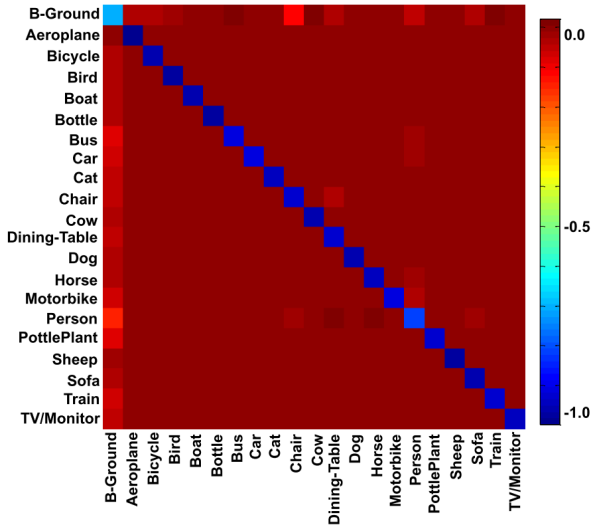


Figure 2-8: Compatibility matrix showing the likelihood of two similar pixels (spatially or appearance wise) being assigned a different class [1].

2-2-3 DeepLab

To achieve better segmentations on a pixel-level, the front-end feature extracting network is of great importance. The features extracted by the front-end are combined in dense layers for classification, or deconvolutional layers for segmentation. With this motivation, the more recent DeepLab network by Chen et al. [22] employs a different front-end, compared to the front-end used in Section 2-2-2. The feature extraction used in DeepLab is called ResNet, presented in He et al. [27]. It alters the straightforward structure, such as the structure in

Figure 2-6, with interconnections between the convolutional layers. This is proven to improve training speed and test accuracy. Another positive feature is that the amount of parameters is not increased in the ResNet model, compared to the networks proposed by Long, Shelhamer, and Darrell [8].

A drawback of segmentation networks is the loss of spatial information during pooling operations. As mentioned in Section 2-1-3, there are multiple approaches to minimizing this negative impact. Figure 2-5 shows multiple solutions, from which atrous convolutions is implemented in DeepLab. Atrous convolutions have the advantage of covering larger patches, without increasing the computational demands. The atrous rate is the amount of space between the considered pixels in the dilated kernel. Figure 2-9 shows what the dilated kernel looks like, compared to a standard kernel. This means that the effect of convolutions on pooled layers can be simulated, while retaining the original resolution. The result is an operation that can capture large scale features without the loss of spatial information.

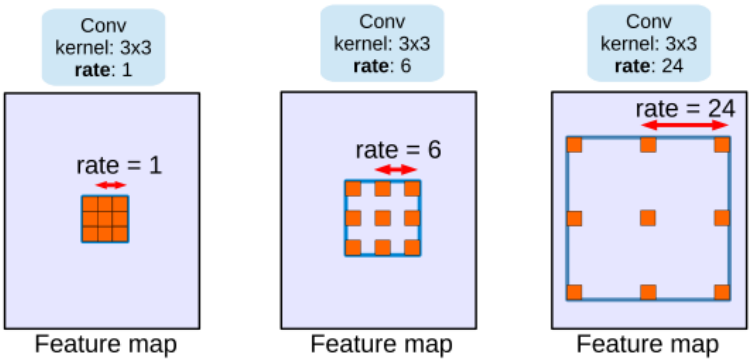


Figure 2-9: A standard convolution kernel (rate=1) compared to dilated, or atrous, convolution kernels. Enlarging the field of view of the kernel allows for capturing image feature information on multiple scales. [12].

To combine the different scale features that can be gathered using different atrous rates, a parallel convolution at different atrous rates is performed. This is referred to as Atrous Spatial Pyramid Pooling (ASPP). Combining atrous convolutions with a pyramid structure results in a feature extraction on multiple scales, without loss of spatial information. With the retained spatial information the reconstructed image has accurate edges. Due to the use of the ASPP, objects at different scales can be detected. The version of DeepLab that is used in this thesis is proposed in Chen et al. [12]. Instead of using a CRF module to use image level features for finer edges [22], the image level features are added to the ASPP. The entire structure of DeepLabv3 [12] is depicted in Figure 2-10. The blocks are based on the structure of ResNet [27].

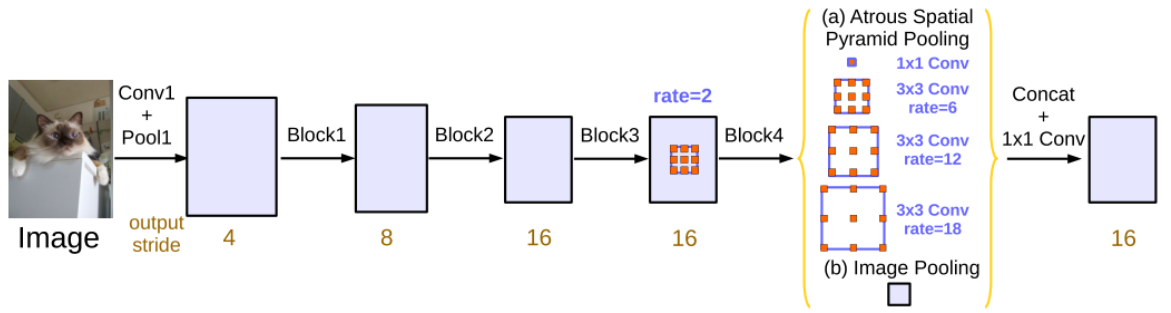


Figure 2-10: The structure of DeepLabv3, containing atrous convolutions and a ASPP. Output stride is the ratio of input resolution to output resolution. In the pyramid pooling module (consisting of multiple layers in parallel), a map with image level features is added to insert accurate spatial information from the original image. [28].

2-2-4 PSPNet

In Zhao et al. [29], Pyramid Scene Parsing is introduced. PSPNet consists roughly of a structure as shown in Figure 2-5, using spatial pyramid pooling. Similar to DeepLab, PSPNet implements a version of ResNet [27] before the pyramid pooling module to construct a feature map. Next, the pyramid module divides the map into multiple pipelines in the network that each operate on another scale. After these separate layers completed their inference procedure, the results are concatenated and the output is produced through a last convolutional layer. A representation of the Pyramid Pooling structure is shown in Figure 2-11.

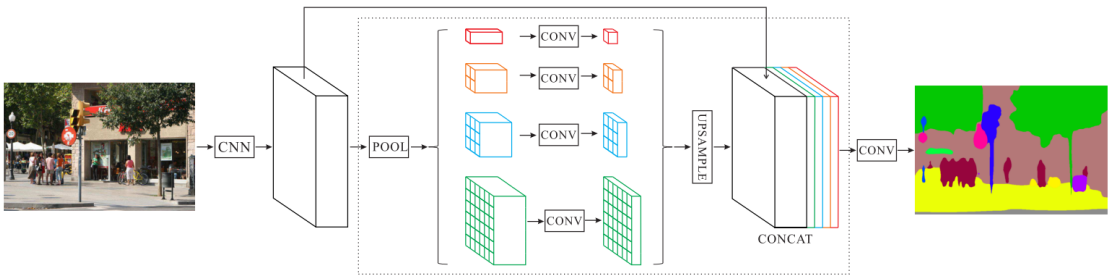


Figure 2-11: Model of the global structure of the Pyramid Scene Parsing Network [29].

An important difference between the pyramid module in PSPNet and the ASPP module in DeepLab is the way in which the multi scale information is gathered. Where DeepLab uses atrous convolutions to cover a larger field of view, PSPNet applies a increasingly large pooling operations to create different scales. PSPNet does not implement any form of image level features to accurately reconstruct edges in the final segmentation.

2-2-5 Mask R-CNN

Mask R-CNN [28] is an addition to Fast R-CNN [30]. These networks distinguish themselves with their region-based approach. A set of bounding boxes is determined, based on their

estimated richness of features [31]. A technique called RoIPool [30] extracts features from all boxes and processes them for classification. In this procedure alterations are made to the features introducing misalignments. This problem does not affect the classification of the ROI, but for pixel-wise classification it is important to retain spatial information in an exact form. A small change to this principle is capable of retaining this information and is called RoIAlign [28]. The distinguishing action between Mask R-CNN and Fast R-CNN is the ability of Mask R-CNN to reconstruct a per-pixel classification for each region of interest.

The mask, or segmentation, is done with a simple approach that upsamples as a branch of the bounding box and classification pipeline (Figure 2-12). While segmentation is the main interest in this project, in He et al. [28] it was merely proposed as an addition to the Fast R-CNN principle. They note that their deconvolutional procedure is rather simple and can be easily extended to increase performance.

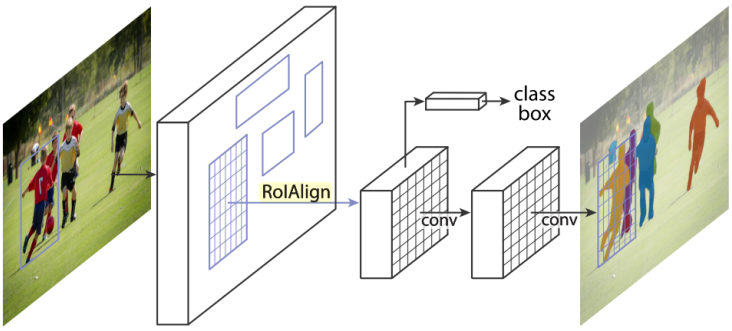


Figure 2-12: Model of the Mask R-CNN from [28].

Chapter 3

Experimental Setup

A Deep Learning pipeline consists of several important parts. To acquire optimal results, factors such as the dataset, data splits, data augmentation and the final evaluation should be carefully chosen. The specific variables used in this work are highlighted in this chapter.

3-1 Datasets

During this project multiple datasets were used to train and evaluate networks. In this section information regarding structure and contents of these sets is provided. Most of the networks presented are used to either pretrain the network. Datasets containing ship specific images are used for fine-tuning a network.

3-1-1 ImageNet

ImageNet [4] is a dataset that kicked off the rapid progress in deep learning for computer vision. The first deep learning solution for image classification that won the ImageNet Large Scale Visual Recognition Challenge (LSVRC) [5] was presented by Krizhevsky, Sutskever, and Hinton [2]. Their network won the challenge in 2012, almost halving the error rate compared to other entries [13]. It is widely used as a performance benchmark in various papers on the task of image classification.

ImageNet contains a vast amount of full resolution images. Over 14 million images are provided with annotations on the contents of each image (in August 2014). The annotation is based on a tree like structure containing super and subclasses. The large amount of images combined with the hierarchical class structure makes ImageNet an ideal dataset to pretrain a Convolutional Neural Network (CNN).

3-1-2 VOC

The Pascal Visual Object Classes (VOC) dataset [32] has been used as a computer vision benchmark since 2005. It is considerably smaller compared to ImageNet, but the VOC set has a different style of annotation. Besides the classification task, a detection task was added to extend the challenge into a new area of computer vision. In 2012 a subset of the VOC dataset was added containing per-pixel annotation with class labels.

Much like the hierarchical structure of ImageNet, VOC is built as a tree of classes with branches and leaves. For example, there is a superclass vehicle, containing boats, trains etc. The 21 classes used in VOC are still used as a standard benchmark for segmentation, for example in Zheng et al. [1].

3-1-3 MSCOCO

An important step in the pixel-level segmentation problem was the introduction of Microsoft Common Objects in Context (MSCOCO) [33]. The size, and therefore the diversity, of the segmentation set from VOC was limited. The amount of images annotated for segmentation tasks in MSCOCO exceeds a hundred thousand, making it exceptionally useful for learning deconvolutional layers in a Fully Convolutional Neural Network (FCN).

Among the datasets mentioned in this section, MSCOCO has the richest annotations. It consists of classification annotation, bounding boxes for detection, masks for segmentation and captions for context in images. The classes covered in this dataset contain all classes from VOC, but added an additional 70 classes. An extra extension the segmentation task was added by discriminating between instances of an object class. This means that two adjacent persons in an image are not annotated as one person patch, but two separate person patches. A drawback in the MSCOCO dataset is the accuracy of the annotation masks. Internally the masks are stored as a collection of key points that can be used to draw the mask. This decreases the annotation file size and allows for overlapping masks. The problem is that the masks are not perfect on a pixel level.

3-1-4 MWC

In cooperation with the Maritime Warfare Centre (MWC), two datasets were devised for this project. At first the goal was to test if the classification system at TNO was able to classify the vessels in the dataset. While this was possible, the segmentation algorithm in place (based on the work of Zheng et al. [1]) suffered from detection failures and poor segmentations. To improve performance a dataset was created suitable for training or fine-tuning a FCN.

Initial Test Set - MWC250

The first set provided was used to test the classification algorithm which was designed to match a ship silhouette to entries in a database. It consists of 250 ship images of varying quality. Each image is provided with the corresponding ship class ID, and a per-pixel segmentation. The segmentations are provided for each image, annotated by three different human operators.

This results in three slightly different segmentations per image. For the purposes of this work we are not interested in these difference, so per-pixel majority vote was applied to construct a single ground truth segmentation map per image. The importance of both the depicted ship class and the corresponding segmentation is further evaluated in Section 3-3.

Improved Set - MWC Large

The ships that are in the images from MSCOCO are not representative for the vessel types this work focuses on. Where MSCOCO has all kinds of rowing boats, it does not contain a large amount of warships. Combining this property with the coarse annotation masks makes MSCOCO a less than ideal choice as a fine-tuning dataset. To train and fine-tune networks on the image domain of warships, the MWC provided a dataset with high-resolution images of warships. Each image is provided with a single segmentation map. There are over 3000 images in the MWC Large set, comparable to the amount of images containing ships in the MSCOCO set. The dataset contains images of ships in multiple situations, lighting conditions and weather situations. However, most images are in good conditions.

Data Split

To test the performance of a network, it is important to have a test subset of your data. This ensures that your performance is not influenced by images that the network was trained on. While most image datasets are very diverse in the type of images they provide, the MWC Large set contains a lot of subsets of images, containing multiple images of the same vessel. Each image is marked with a unique id tied to the ship in the image. Splitting the dataset based on this id would ensure that one ship in slightly different orientations occurs in both the training and test set. A problem with a split based on the ship id is that multiple ships can be part of the same class, meaning that they are of identical shape. To avoid this, a split was made based on class names. The resulting data split sizes, based on vessel classes, is shown in Table 3-1.

Table 3-1: The amount of images in each split of the dataset.

Data split	Amount of Images
Train set	2397
Validation set	304
Test set	302

3-2 Noise Model

In most datasets the input images are of decent quality. This is ideal to train a network with the full potential of the information contained in an image. For our application, it is expected that during operational use the images presented will not be of the high quality presented in datasets such as MWC-Large. To increase performance on images that are disturbed by noise, we draw inspiration from the approach taken by Tremblay et al. [34]. The data is augmented with disturbances similar to disturbances that are observed during

operation. This will force the network to learn features that are present in both high and low noise situations. Furthermore, features that are only present in low noise images are not considered by the network in this setup. The performance is expected to be more robust to noise, but exhibit decreased performance in low noise images due to the absence of the features learned from clean images. To research the presence and influence of this trade-off, we train our network on a range of noise levels as presented in Section 3-2-2. The design of the noise model is presented in Section 3-2-1.

3-2-1 Noise Model Structure

Multiple disturbances can occur during capture of an image. The resulting degradations of the image are presented as a contrast reduction, a color shift, a blurring effect, and image noise. The sequence of application of these disturbances is visualized in Figure 3-4.

Image Blur

The first disturbance introduced is blur. During imaging a variety of phenomena can introduce a blurring effect [35]. For simplicity we only implement a single blurring operation to cover all these phenomena. An example of blur on an image is given in Figure 3-1. The blurring effect is achieved by convolving the input image with a one-dimensional Gaussian kernel in the horizontal direction, followed by a convolution in the vertical direction. The Gaussian kernel is given in Eq. (3-1). Here the parameter x is the distance to the center of the kernel and σ_{blur} the standard deviation. The size of the kernel is governed by the standard deviation σ. It is cut off at 4σ from the center at each side of the kernel. The edges of the image are extended by mirroring the contents of the image, extending the image to allow the kernel to calculate values at the image edges.

$$G(x) = \frac{1}{\sqrt{2\pi\sigma_{blur}^2}} \exp\left(-\frac{x^2}{2\sigma_{blur}^2}\right) \tag{3-1}$$

(a) Input image. **(b)** Blurred image.

Figure 3-1: The effect of image blur applied with a filter kernel shown in Equation 3-1. The σ_{blur} value is 1.5. Note the loss of color features during the application of the blurring operator.

Contrast Reduction

The second disturbance is a contrast reduction, combined with a color shift. The contrast reduction and color shift are an effect from scatter in atmospheric aerosols such as fog and vapor [36]. To mimic this disturbance, a solid color is merged with the original image to reduce the contrast and introduce a color shift. The equation governing this disturbance is shown in Eq. (3-2). Here, λ is the parameter that indicates the intensity of the contrast reduction. Where 0 is no reduction and 1 is fully shifted to a solid color. The parameter I_{color} is a uniform randomly sampled Red Green Blue (RGB) color. An example of this operation is depicted in Figure 3-2. Based on the work of Tremblay et al. [34], there is no need to sample from naturally occurring colors. By including all colors we try to learn invariance with respect to the color shift.

$$I_{disturbed} = (1 - \lambda)I_{input} + \lambda I_{color} \tag{3-2}$$

(a) Input image. **(b)** The solid color used for merging. **(c)** Contrast reduced image, with a color shift.

Figure 3-2: The effect of contrast reduction and color shift, according to Equation 3-2. The λ value is 0.7. The RGB values of the gray solid color are [0.8,0.8,0.8].

Sensor Noise

There are multiple sensor related noise sources present during digital image capture, as described by Nakamura [37]. There are noise sources that have a fixed influence, and noise sources that are temporal of nature. Two often occurring temporal noise types are thermal noise and shot noise. Thermal noise arises from the thermal agitation in resistors, resulting in unpredictable signals that flow through these resistors. Shot noise is the result of an inconsistent amount of photons reaching the sensor during a single capture cycle. While all noise types have their own typical form of noise behavior, this model implements additive Gaussian noise per pixel. The mean of the noise is zero, and the standard deviation of the noise (σ_{sensor}) is varied at multiple noise levels, as described in Section 3-2-2. With that configuration, an invariance towards multiple types of noise is learned by the network. An example of the additive Gaussian noise added to an input image is shown in Figure 3-3

(a) Input image. **(b)** Sensor noisy image.

Figure 3-3: The effect of Gaussian noise applied with a mean value of 0 and σ_{sensor} value of 50. This is an extreme value to visualize the influence of this disturbance.

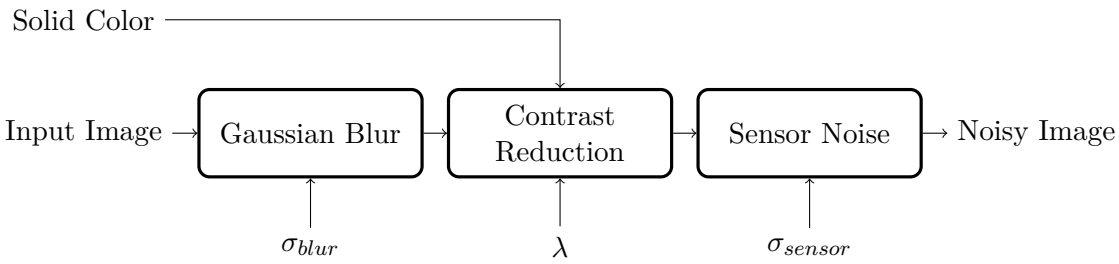


Figure 3-4: The noise model pipeline.

3-2-2 Model Parameters

The influences of increasing noise on the training data of the network is explored by creating multiple datasets. The datasets are augmented with noise through the pipeline presented in Figure 3-4. To test the performance of the networks trained on noisy data, a test set is created for each noise level. The parameters of the noise model are uniformly sampled from intervals. The interval values increase with each noise level. The value intervals are shown in Table 3-2. Examples of resulting images are shown in Figure 3-5.

Table 3-2: The parameter intervals from which the noise levels are defined.

Noise Level	λ	σ_{blur}	σ_{noise}
0	[0,0]	[0,0]	[0,0]
1	[0.1,0.2]	[0.3,0.6]	[0.0,2.0]
2	[0.2,0.4]	[0.6,1.0]	[2.0,4.0]
3	[0.4,0.6]	[1.0,1.5]	[4.0,5.5]
4	[0.6,0.8]	[1.5,1.9]	[5.5,7.0]
5	[0.8,0.83]	[1.9,2.1]	[7.0,9.0]
6	[0.83,0.86]	[2.1,2.3]	[9.0,11.0]

(a) Noise level 0

(b) Noise level 1

(c) Noise level 2

(d) Noise level 3

(e) Noise level 4

(f) Noise level 5

(g) Noise level 6

Figure 3-5: Noise augmented images based off the intervals provided in Table 3-2, and the noise model in Figure 3-4. The color shifts shown are not noise level specific, but randomly sampled per image from RGB color space.

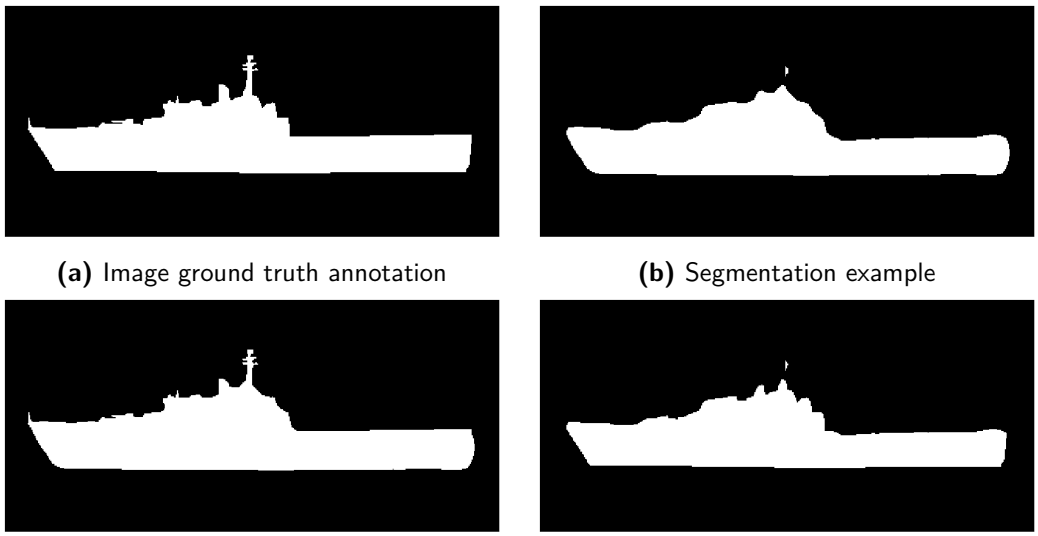
3-3 Performance Metrics

To evaluate whether the outcome of a network is of high quality or not, appropriate performance metrics should be defined. An adaption of the Intersection over Union (IoU) presented in Long, Shelhamer, and Darrell [8] is used to assess the quality of the segmentation, compared to the ground truth. The final classification of the segmented vessel is of high importance to the project. Therefore, the ranking returned by the classification algorithm on the MWC250 set is used to assess the performance of the network segmentations during classification. For the test dataset from MWC Large, there are no ship class ground truths available. This means that there is no way to retrieve the rank of the class prediction. To account for this, a metric that compares rankings is used to assess classification performance.

3-3-1 Intersection over Union

In Long, Shelhamer, and Darrell [8] multiple metrics for evaluation are presented. The metric used in multiple segmentation challenges is the mean IoU. For this application a slight variation of this metric is used. Instead of taking the mean over the IoU of each separate class, only the ship class is considered. The only remaining class considered in this project is the background. Including the score for the background would increase the IoU score, because even with a poor ship segmentation, there will be a large amount of pixels correctly classified as background. The goal is to have a rating of 0 when there is no ship detected, and a rating of 1 for a pixel perfect segmentation. This is achieved when only the IoU of the ship class is considered. In Eq. (3-3) the metric is described. The terms TP,FP,FN are True Positive, False Positive, and False Negative respectively. A True Positive is a correctly predicted ship label, a False Positive is an incorrectly predicted ship label. A False Negative is a ship label that is predicted as background. There is also a True Negative, but as mentioned earlier, these are not considered in this metric. In Figure 3-6 the union and overlap are visualized in an example.

$$IoU = \frac{\text{Area of Intersection}}{\text{Area of Union}} = \frac{TP}{TP + FP + FN} \tag{3-3}$$



(a) Image ground truth annotation

(b) Segmentation example

(c) The union of (a) and (b)

(d) The intersection of (a) and (b)

Figure 3-6: Visualization of the IoU metric with a . According to Eq. (3-3) the IoU is 0.887

3-3-2 SACEI Classification Score

The IoU metric is effective on judging the quality of the segmentation. However, not every pixel in the final segmentation is of equal importance during the classification procedure. This difference is discarded in the IoU metric. The unknown influence of different pixels means that a high IoU metric is no guarantee of a successful classification. During the classification, the segmentation of the vessel is scaled to be 100 pixels wide. Pixels at the bow and stern of the vessel will therefore influence the produced scaled segmentation heavily.

When presenting the SACEI classification system with a segmentation, a ranking of database entries is returned. These entries are ranked in order of a distance metric. The distance is the summation of the absolute per-pixel difference in value between the scaled segmentation, and all database entries. For the dataset MWC250, a set of ship class ground truths is provided. With the ranking and the class ground truth, the classification score can be derived. The rank at which the ground truth class appears, is the resulting classification score.

A feature of this classification method is that it does not require a ground truth segmentation to perform. A drawback is that the classification algorithm does not return a perfect score for all human annotated segmentations. To account for this, the human annotated segmentation masks are evaluated using this method as well. The result of the classification procedure on the ground truth segmentations provides us with a benchmark for human performance. We can use this human performance score to assess the quality of the score achieved with automated approaches, such as the networks in Section 2-2.

3-3-3 Rank-Biased Overlap

A solution must be found for datasets that do not contain a class ground truth. Without the ground truth, the classification score cannot be derived. This makes the ranking returned by

the SACEI classification algorithm useless. Using the Rank-Biased Overlap (RBO) algorithm as proposed in Webber, Moffat, and Zobel [38], two rankings can compared to each other, returning a score based on their similarity. This is useful for the dataset MWC Large, that does not contain class ground truth labels for each image. By evaluating the relation between the classification score and the RBO, it is possible to assess the quality of the SACEI ranking without a class ground truth label.

A useful feature of the RBO algorithm is a weighting value p. This value is used to tune the algorithm in its focus on higher ranking results and their influence on the resulting RBO score. For example, a low p value will make the algorithm top-weighted, meaning entries at the top of the compared rankings have the most influence on the final score. Increasing p towards 1 will increase the depth of comparison, taking into account entries lower in the ranking. This is very useful for our purposes, because we are most likely only interested in the similarity at the top of the returned ranking by the SACEI algorithm.

The behavior of the RBO score is explored in Figure 3-7. In this figure a test list is compared to a reference list. Both originally consist of an ordering of 1 to 100. The top n entries of the test list are shuffled, with n shown on the horizontal axis. This means for example that at n=10, the top 10 of the test list is shuffled randomly, while the rest of the list remains unaltered. To account for the random behavior of the random shuffling, each n-value is shuffled and examined 100 times. Subsequently, the mean is evaluated.

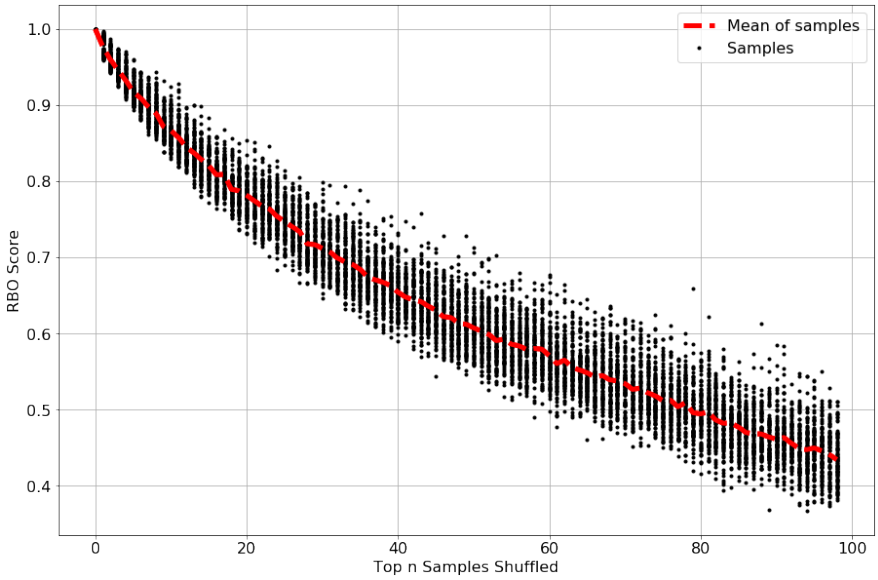


Figure 3-7: Behavior of the RBO score, while randomly shuffling the top n entries in a list of 1 to 100. Shuffling increasingly more entries, decreases the RBO score compared to a list of 1 to 100. A value of $p = 0.98$ was used.

Chapter 4

Experiments

In Chapter 2 different networks are discussed. Combined with the tools from Chapter 3, we can start evaluating the performance of these networks. A benchmark performance is done on the MWC250 dataset to assess the initial segmentation performance of different networks. The networks that show the most potential are used to move on to the MWC Large dataset. Applying this data in a training configuration is hypothesized to improve performance. The resulting trained networks are applied to a test set, and the generated segmentations are evaluated.

4-1 Algorithm Selection

Based on Chapter 2, four networks are selected to test on the MWC250 set as a benchmark. Based on their reported performance in literature, and their inherent differences in structure, these networks provide an insight into the best configuration for our application. The networks are used without any training done on our datasets, to assess their performance based on the weights provided. Evaluation is done on network scaling performance, SACEI ranking, Intersection over Union (IoU) score, and the detection failure rate.

4-1-1 Input Image Cropping

Due to the input size constraints of multiple networks, input images are scaled down to a 500x500 pixels size. When there is an area surrounding the vessel, or Region of Interest (ROI), this reduction in resolution can result in a very small ROI to detect. Being able to detect both large and small objects in an image is referred to as the scale-invariance of a network. Not all networks have the same properties in this domain. This means that some networks are strong in detecting small objects, and some are not. Some networks are even worse at detecting large objects compared to smaller objects. To investigate the influence of the size of the ROI in the image, all experiments in this section are run on the original image, as well as on a cropped image containing only the ROI. An example of this conversion is shown in Figure 4-1.

(a) Original image. **(b)** Cropped to ROI.

Figure 4-1: An illustration of the cropping procedure, converting an image into a ROI only image.

4-1-2 Visual Comparison

Due to the different structure of the networks applied in the benchmark in this section, the resulting segmentations are different as well. Deconvolutional layers that upsample the final segmentation map are combined and structured in a variety of ways. This has an influence on their ability to reconstruct edges in the final segmentation. Poor edges can be a result of applying a simple network structure as proposed in Long, Shelhamer, and Darrell [8], without any connections between the first and last layers. In Zheng et al. [1] a Conditional Random Field (CRF) based module is applied to reconstruct edge information. Another solution is to avoid pooling with the use of atrous convolutions [12]. An overview of sample segmentations is provided in Figure 4-2.

It is immediately visible that there are big differences in the segmentation results. The CRF-RNN segmentation in Figure 4-2c stands out because of its fine edges. This is a direct result of the CRF module that incorporates the information in the original images, before spatial information is lost due to pooling. In Figure 4-2d the rough outlines of the hull of the ship are detected. All other elements are not reconstructed. The downside is that these omitted elements are a crucial factor in the classification process. The segmentation in Figure 4-2e is somewhat better, but still lacks detail at the edges. The last segmentation, shown in Figure 4-2f, is of quite high detail. It does not omit crucial fine edge features, but is not as accurate as Figure 4-2c. Based on this first look at the different segmentation results, CRF-RNN and DeepLab seem to have the most promising result.

4-1-3 SACEI Ranking

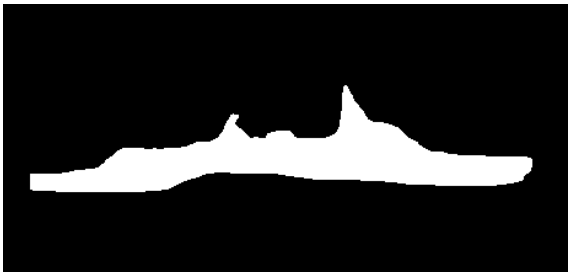
While the IoU score is a good indicator of segmentation performance, it is not straightforwardly connected to a high SACEI classification rank. This is mentioned in Section 3-3-2. The ranking returned by the classification algorithm is influenced by multiple hard to quantify factors, such as the uniqueness of the silhouette and accuracy of the database entries. This makes it hard to pinpoint the direct cause of failing classifications on segmentations with a high IoU. Nevertheless, the final outcome of the classification algorithm is of great

(a) Original image.

(b) The ground truth annotation.

(c) CRF-RNN segmentation.

(d) Mask R-CNN segmentation.



(e) PSPNet segmentation.

(f) DeepLab segmentation.

Figure 4-2: Sample segmentations resulting from the networks mentioned in Section 2-2.

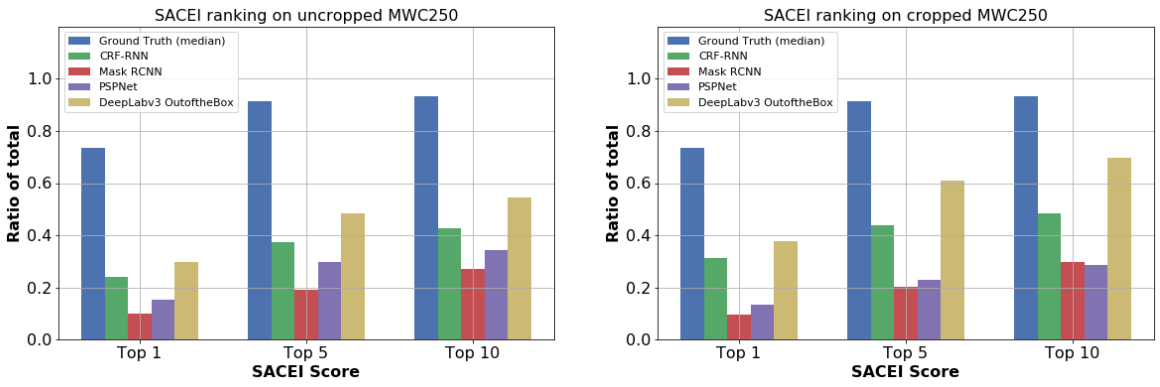
importance to the project. The significance of the SACEI classification score is of greater importance compared to the IoU score, in terms of value to the SACEI project.

The results of the initial four networks are depicted in Figure 4-3. The score of the human annotated ground truth set is included to analyze the performance of the classification algorithm. A Top 1 score indicates that the classification algorithm assigned the correct class label as number 1 in the ranking. Top 5 indicates the correct label is among the top 5 in the predicted labels. The Top 1 scores are summarized in Table 4-1. In line with the findings in Section 4-1-2, CRF-RNN and DeepLab are the highest scoring networks in their initial configuration. A notable result that is not in line with the findings in Section 4-1-2, is DeepLab outperforming CRF-RNN. Even while the edge reconstruction capabilities of CRF-RNN are considerably more accurate. In Section 4-1-4 the reason for this result is explored.

An interesting result visible in Figure 4-3 and Table 4-1 is the impact of cropping input images. Both DeepLab and CRF-RNN exhibit an increase in performance when the input image is cropped. Mask R-CNN does not seem to be affected by the difference, which can be explained by the low edge quality. This network, as depicted in Figure 4-2d, tends to ignore finer features at the edge. Slightly enlarging the features has no effect on the final result. This is further explored in Section 4-1-5. PSPNet performance decreases when cropping the input. This is a result of the structure of PSPNet, that explicitly focuses on smaller scale objects. This makes PSPNet a good candidate for segmenting cluttered scenes with small objects, but a bad candidate for accurately segmenting a single large object.

Table 4-1: SACEI Top 1 scores per network, for both cropped and original images. Best performance in bold (excluding the ground truth score).

Network	Original	Cropped
Ground Truth	0.7344	0.7344
CRF-RNN	0.2415	0.3112
Mask R-CNN	0.0996	0.0954
PSPNet	0.1535	0.1328
DeepLab	**0.2988**	**0.3776**



(a) Original MWC250 images.

(b) Cropped to ROI MWC250 images.

Figure 4-3: SACEI classification results from the different networks presented in Section 2-2. For reference, the performance of the ground truth segmentation maps is included.

4-1-4 Detection Failure Rate

Poor segmentations that contain only rough edges of a vessel are able to score high in the SACEI ranking. Some vessels are more unique in their silhouette feature, making them easily distinguishable. This allows for an inaccurate segmentation with a low IoU score to return a high rank in the classification step. In other words, a slightly lower IoU score is not a guarantee for failure. However, some networks are sensitive to disturbances and noise. Depending on the network, the possibility exists that there is no vessel detection. Such a detection failure results in a IoU score of 0, and a classification step is skipped. Investigating the rate of a detection failure gives an insight into the robustness of the network in the original weight configuration.

The detection failure rates are depicted in Figure 4-4. The most influenced network is CRF-RNN, almost halving the detection failures when cropping the input images. As previously mentioned in Section 4-1-3, PSPNet is not designed for this type of segmentation scenes. The increase in detection failures for cropped images therefore is not a surprise. The other networks do improve, but only slightly.

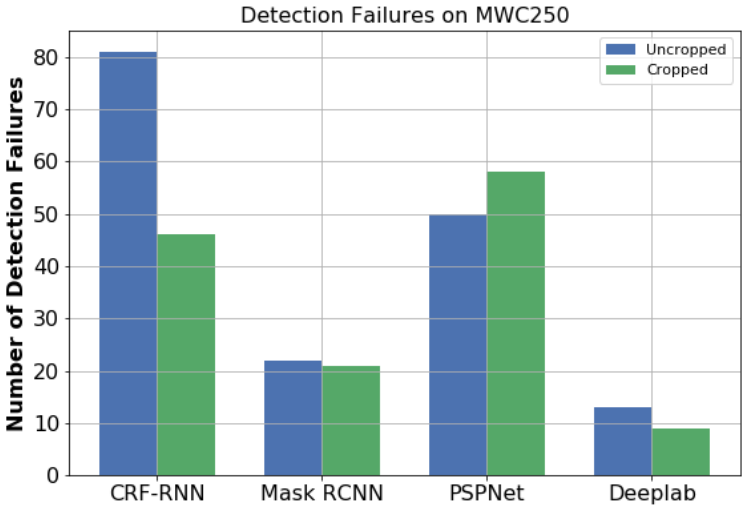


Figure 4-4: Visualization of the detection failure rate. Cropped images are the originals with background surrounding the ROI removed.

4-1-5 IoU Score

A more straightforward metric, compared to the SACEI ranking, is the IoU score. This score is introduced in Section 3-3-1. In Figure 4-5 both the score derived after inference on the cropped and the original images is shown. Mask R-CNN and DeepLab are the best performing networks, which is most likely due to their low detection failure rate (Figure 4-4). The changes in performance that arise due to the cropping of the images are in line with the performance differences in Table 4-1. This gives an indication that there might be some correlation between the IoU and the SACEI rank.

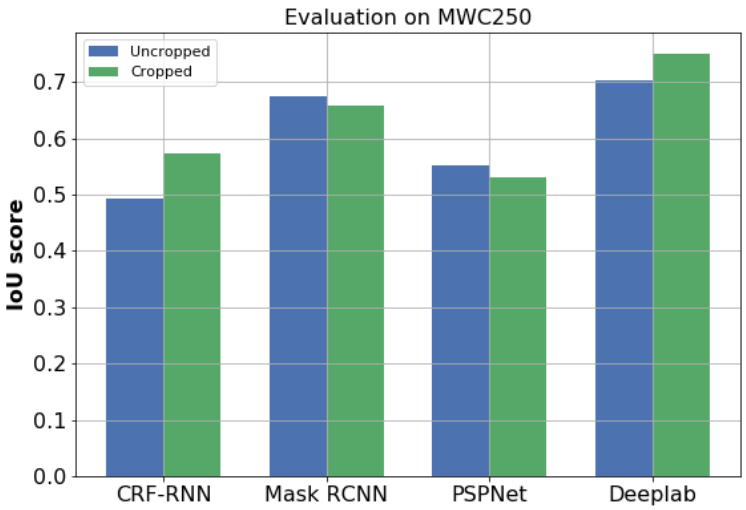


Figure 4-5: The performance of separate networks with their initial weight set. The IoU score is based on Eq. (3-3).

4-1-6 Conclusions on the MWC250 Benchmark

Based on the evaluations in the previous sections, a choice is made to proceed with experiments with only one out of the four networks presented. The performance with the weight sets provided is used to assess which networks show the most promise. Based on the IoU score, Mask R-CNN and DeepLab are the highest scoring networks. The negative effects of detection failures is embedded in this scoring. As mentioned in Section 3-3-2, this is not a guarantee for a high rank in the SACEI classification. Based on the results of the experiment on that ranking, CRF-RNN and DeepLab are the preferred choices. This clearly indicates that there is a possibility of scoring high on the IoU metric, but poorly in the SACEI ranking. Based on the visual results in Figure 4-2, Mask R-CNN covers a large area of each vessel. Put together with the low detection failure rate adds up to a high IoU score. However, the Mask R-CNN results in Figure 4-2d show a lack of fine features along the edge of the segmentation. These features are crucial to the SACEI classification algorithm, as is clear from the low SACEI ranking achieved by the Mask R-CNN.

As previously mentioned, the SACEI ranking is the leading metric for performance. DeepLab is the network chosen to proceed with, based on the promising results of DeepLab. Based on the high performance of DeepLab regarding detection failures, DeepLab will be used to explore the effects of training on domain specific data, as well as training on noise augmented data.

4-2 Fine-Tuning DeepLab

The dataset presented in Section 3-1-4, provided by the Maritime Warfare Centre (MWC), opens up a range of possibilities for improving current networks. Using the high quality images with accurate annotation, we can fine-tune the DeepLab network. We evaluate the influence of initial weight sets, and decide which point is most suitable for our purposes. Subsequently, we apply the noise model presented in Section 3-2. The noise model is used both in evaluating noise robustness in the DeepLab network, and in the effects in terms of noise robustness after training on noise augmented data.

In Table 4-2 the parameters used to obtain the results in this section are presented. Training the batch normalization parameters requires a large batch size [12], which was not possible due to limited memory on the hardware used for experiments. The framework was implemented in TensorFlow [39].

Table 4-2: Training parameters used in the DeepLab training routine.

Parameter	Value
Learning rate	0.001
Atrous rates	[6,12,18]
Batch size	12
Training iterations	10,000
Fine tune batch normalization	False

4-2-1 Initial Point of Training

To assess if the pretrained weights provided with the DeepLab model are a suitable initial training point, we compare the IoU results of different DeepLab configurations. First, we evaluate the initial DeepLab. Secondly, we evaluate the DeepLab network fine-tuned on our dataset. Lastly, we train DeepLab with the weights initialized on a checkpoint pretrained on ImageNet. The resulting scores are presented in Table 4-3.

Table 4-3: IoU evaluation scores from different DeepLab models. Evaluated on the MWC Large test set. Best performance is in bold text

DeepLab Network	IoU Score
Trained from ImageNet	0.862
Trained from provided weight set	**0.890**
No training, provided weight set	0.828

Based on the results in Table 4-3 we can conclude that fine-tuning on our two class dataset improves the segmentation performance. Training DeepLab on MWC Large improves the performance, regardless of the initial weight distribution. Training from the weight checkpoint provided with the DeepLab model is shown to be performing better compared to the version trained from the ImageNet checkpoint. As previously mentioned, hardware constraints keep us from being able to fine-tune batch normalizations parameters. While these are pretrained in the provided checkpoint, they are not pretrained in the ImageNet initial weights. This is one

difference between the two initial points that explains the observed performance difference. The high performance of the DeepLab version trained with the provided weight set, makes the provided weight set the preferred starting point for subsequent training experiments.

4-2-2 Noise Data Augmentation

As proposed in Section 3-2, we will augment our data (MWC Large, Section 3-1-4) with a noise model. The noise model will simulate natural phenomena that disturb an image, both before and during capturing a digital image. In this experiment we explore both the performance over all different noise configurations, as well as the performance after training on every noise configuration. We expect that the network trained on a clean dataset (Noise level 0 in Table 3-2) will perform very well on low noise levels. The higher the noise level gets, the worse the performance will be. Networks that are trained on data that is already noisy are expected to be more invariant to the disturbances. This assumption is based on the work of Tremblay et al. [34], where non-realistic data augmentation is used to force a network to focus on different features during training. Besides the heightened invariance to noise, it is expected that networks trained on noise have lower performance on the clean dataset. This is expected due to the fact that in these cases the test set domain is further removed from the training images, compared with a network that is trained and tested on the same domain.

Noise Application

In order to be able to use the network training routines in an identical manner compared to the training on the clean MWC Large data, the training data is augmented off-line. This results in six extra data sets, with increasing noise levels. It also means that the data is augmented permanently, instead of a different augmentation each time it is presented to the network for training. One of the benefits of data augmentation is that you can artificially enlarge your dataset, but we just change the input data. However, as shown in Table 3-2, we sample parameters from an interval. This means that if we duplicate the dataset multiple times, the result will always be different. We decided to compromise between the diversity benefits of having an on-line noise generator, and the implementation benefits of an off-line version. The entire dataset is augmented with the noise generator, and this is done five times. The resulting sizes of the dataset splits are shown in Table 4-4.

Table 4-4: The amount of images in each split of the dataset after noise augmentation.

Data split	Amount of Images
Train set	11985
Validation set	1520
Test set	1510

Results from Noise Augmentation

A DeepLab network is trained using the parameters from Table 4-2. The dataset MWC Large, provided by the MWC, is augmented with multiple noise levels (Section 3-2). This results

in seven datasets, ranging from the original images to high noise augmented data. The noise augmented dataset is comprised of three splits, as shown in Table 4-4. The values for the original dataset are shown in Table 3-1. Training the DeepLab model from the initial point chosen in Section 4-2-1 on different noise levels results in eight different networks. These networks are shown in the first column of Table 4-5. This set of networks is comprised of the original model, and all models resulting from training on the MWC Large set. In Table 4-5 each row represents a network. Each column is an evaluation score on a test-split of the MWC Large set in each column, with varying noise levels.

A certain trend is visible in Table 4-5. This trend is visualized in Figure 4-6. Due to the increasing disturbances in the evaluation images, IoU performances drop. Each network exhibits a different curve, indicating varying robustness to the noise levels. An increasing robustness to noise can be seen when a network is trained on higher noise levels. However, a trade off with the performance at low noise levels is present. To assess the performance on the SACEI classification algorithm, the IoU scores relation to the SACEI ranking is investigated in Section 4-3.

Table 4-5: The IoU score of networks trained on different noise levels, and subsequently evaluated on different noise levels. Columns represent different evaluation sets. Best performances are bold.

	Test sets						
Network & Training	NOISE0	NOISE1	NOISE2	NOISE3	NOISE4	NOISE5	NOISE6
DeepLab no training	0.8278	0.8240	0.8145	0.7545	0.4099	0.0345	0.0016
DeepLab Noise 0	0.8900	0.8882	0.8804	0.8499	0.6971	0.2642	0.0499
DeepLab Noise 1	**0.8916**	**0.8905**	0.8846	0.8580	0.7340	0.3534	0.0752
DeepLab Noise 2	0.8882	0.8885	**0.8861**	0.8703	0.7951	0.4772	0.1257
DeepLab Noise 3	0.8863	0.8852	0.8830	**0.8747**	0.8454	0.7426	0.5120
DeepLab Noise 4	0.8733	0.8735	0.8726	0.8688	**0.8550**	0.8303	0.7923
DeepLab Noise 5	0.8512	0.8513	0.8499	0.8530	0.8519	**0.8412**	**0.8254**
DeepLab Noise 6	0.8476	0.8477	0.8421	0.8426	0.8413	0.8354	0.8253

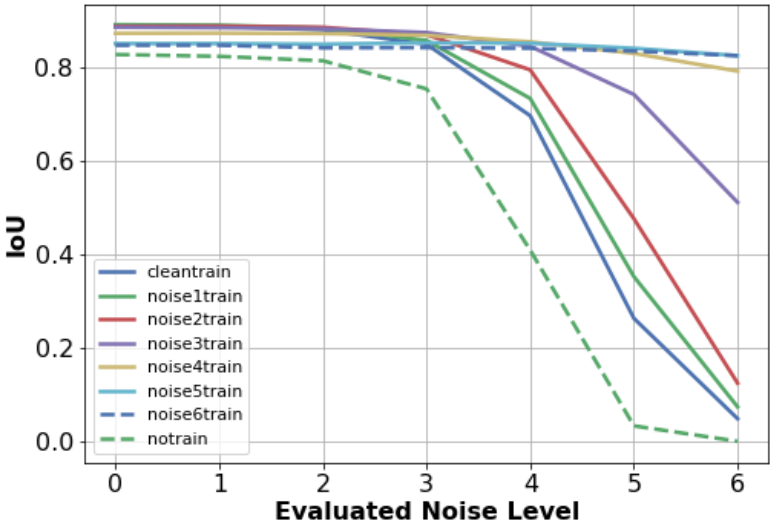


Figure 4-6: The performance of differently trained DeepLab networks, plotted against the increasing noise levels.

4-3 Noise Augmented Networks on MWC250

In Section 4-2-2 we showed that introducing noise to the data results in a network that is more robust to noisy test data, regarding the IoU metric (Section 3-3-1). However, as suggested in Section 3-3-2, there is no linear relation between the IoU metric and the SACEI rank. To investigate whether IoU can be used to estimate SACEI ranking, we compare the two results on the MWC250 set. This dataset contains both the ground truth for the segmentation and the ship class. Subsequently, we evaluate if the Rank-Biased Overlap (RBO) metric from Section 3-3-3 is a good estimator for the SACEI rank.

4-3-1 IoU and SACEI Scores

In this section we investigate a possible relation between the IoU and the SACEI rank. Given a relation, it will be possible to give an estimate on the SACEI rank, based on the IoU score. The advantage would be that there is no longer a need for a ship class ground truth to estimate the classification performance. Secondly, the evaluation of the IoU metric is much faster compared to computing the SACEI classification ranking.

The networks trained in Section 4-2-2 are used to evaluate the MWC250 dataset, yielding IoU scores. The results are given in Table 4-6 and depicted in Figure 4-7. Regarding the results in Table 4-6, there is a discrepancy when compared to the scores in Table 4-5. The IoU results reported on the MWC250 set are not in line with the results on the original MWC Large dataset. This is due to the fact that the MWC250 dataset contains noisy images, as well as clean images. For this particular mix of noisy data and clean data, a DeepLab network trained on MWC Large images of Noise level 3 yields the highest IoU performance. Figure 4-7

shows that the performance starts to drop for networks trained on noise levels 4 and up. The performance deterioration at noise level 4 implies that the MWC250 set does not contain enough high noise images to compensate for the lower accuracy at low noise levels. These values are summarized in Table 4-6.

Table 4-6: The IoU score and SACEI Top 1 rate of networks trained on different noise levels, and subsequently evaluated on the MWC250 dataset. Best performance is bold.

Network & Training	IoU	SACEI Top 1 Ratio	SACEI Top 5 Ratio
DeepLab no training	0.7788	0.378	0.610
DeepLab Noise 0	0.8809	0.656	0.830
DeepLab Noise 1	0.8842	0.618	0.838
DeepLab Noise 2	0.8866	0.639	0.822
DeepLab Noise 3	**0.8873**	0.635	0.846
DeepLab Noise 4	0.8712	**0.660**	**0.855**
DeepLab Noise 5	0.8505	0.610	0.809
DeepLab Noise 6	0.8487	0.552	0.805

Based on the results from Figure 4-7, it is expected that DeepLab trained on Noise Level 3 data will yield the best SACEI scores. However, the assumption that there is a direct relation between IoU and SACEI score does not hold. The SACEI score rates are depicted in Figure 4-8 and summarized in Table 4-6. Here we can see that networks trained on noise levels 0 and 4 outperform the networks trained on levels 1,2 and 3 in the SACEI ranking, even though it is the other way around on the IoU score. To investigate the relation between the IoU and the SACEI rank we investigate individual samples in the next section.

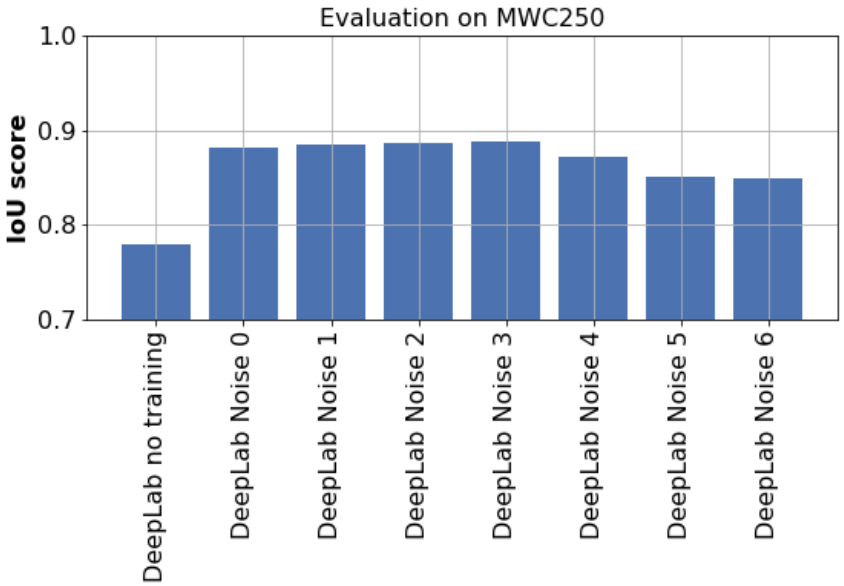


Figure 4-7: The IoU scores on differently trained DeepLab networks.

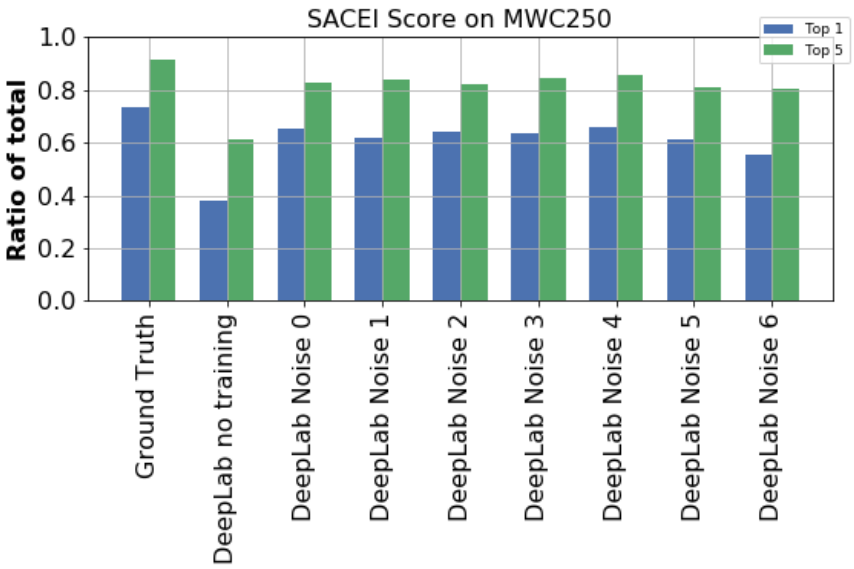
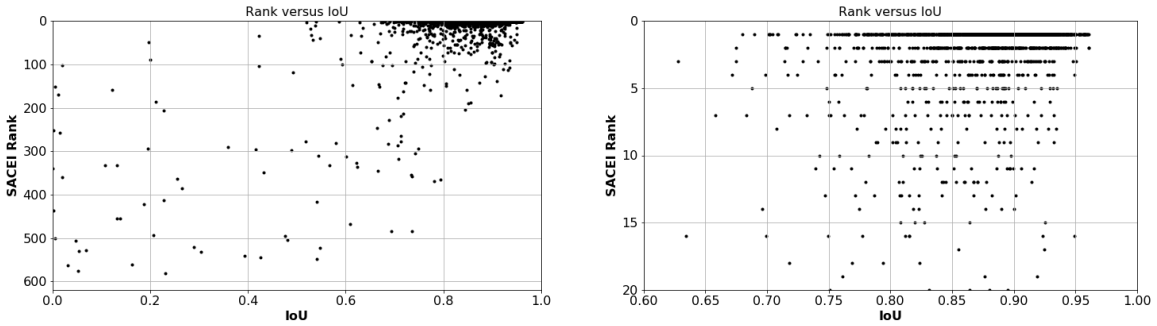


Figure 4-8: The SACEI scores on differently trained DeepLab networks.

Single Sample Analysis

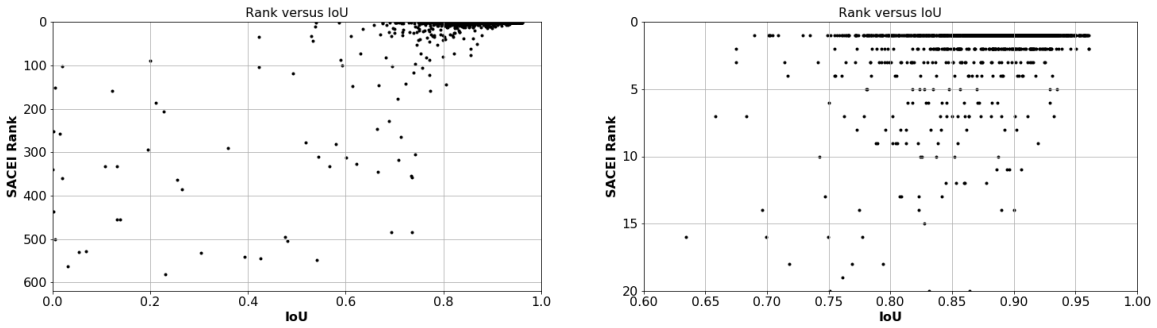
To analyze the relation between the IoU and SACEI scores, we visualize each individual result as a scatter of the IoU versus SACEI rank. The result is shown in Figure 4-9. Something that is immediately visible is the amount of high IoU scores that are paired with a low SACEI rank. This is due to the samples in the dataset that, even with human annotated contours, are not ranked correctly in the classification algorithm. This leads to samples with a high quality segmentation that are not ranked on a high position by the SACEI algorithm. To attune for this contamination of the relation between the two metrics, we filter out all samples that did not rank Top 1 with the ground truth annotation. These results are depicted in Figure 4-10. In Figure 4-10a, it is visible that low rankings are reduced in high IoU ranges.



(a) All results. **(b)** Zoomed in on the cluster.

Figure 4-9: The IoU versus SACEI results of each of the networks.

When zooming in on the cluster, shown in Figure 4-10b, we see multiple interesting things. The first is that the Top 1 ranked samples are wide spread. This supports the idea that

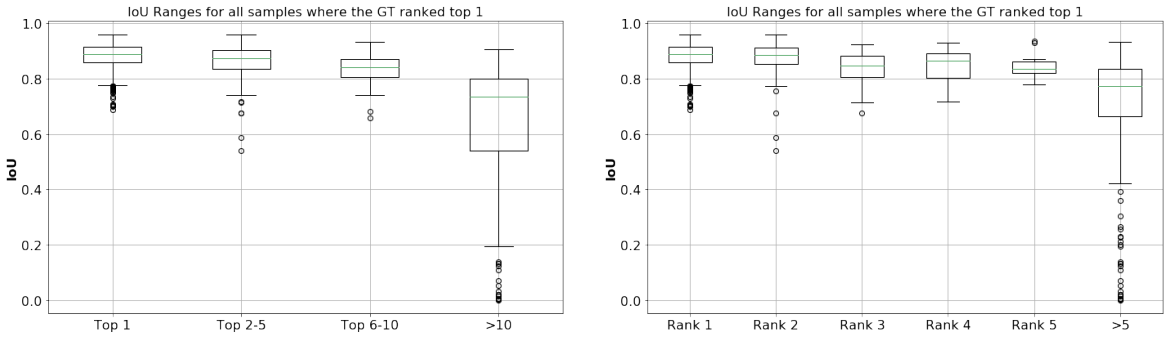


(a) All results. **(b)** Zoomed in on the cluster.

Figure 4-10: The IoU versus SACEI results of each of the networks. Here all samples that did not have a Top 1 score for the corresponding ground truth segmentation are omitted.

high SACEI ranks are not necessarily dependent on extreme IoU scores. It strengthens the assumption that the different pixels have a varying influence on the resulting SACEI score, but a constant influence on the IoU score. This creates a gap where the IoU score is not sufficient to accurately predict a SACEI Rank. This is also visible by the second observation of Figure 4-10b. The IoU scores that rank top 1 are not much higher compared to the lower ranking IoU scores. In Figure 4-11a it is shown that the overlap is very high between the different rank-intervals. Zooming in on the ranks, in Figure 4-11b, we see that the mean values tend to decrease. However, the overlap in the distribution is still high. The amount of samples per rank rapidly decreases after the top 1 ranking. This implies that a high IoU gives a higher chance on a top 1 ranking from the SACEI algorithm, but is certainly no guarantee. Based on the visualizations in Figure 4-11, we can assume that IoU scores lower than 0.7 will not result in an accurate SACEI classification.

To assess the difference between the samples that rank Top 1 and Top 2 (for example), we evaluate their distribution characteristics. If there are significant differences between two rank samples, we can assume that the ranking difference is partly caused by the difference in IoU score. We use the Kolmogorov-Smirnov (KS) test (two-sample) to determine whether both Top 1 and Top 2 scores are pulled from the same IoU distribution. In Table 4-7 the results are presented. The higher the KS-statistic, the smaller the confidence that two distributions are samples from a single parent distribution. Similarly, the lower the p-value, the lower the confidence that two distributions are samples from a single parent distribution. From Figure 4-11b we can already assume that samples ranking 10+ in the SACEI classification algorithm have a significantly different distribution compared to the Top 1 ranking samples. This is confirmed in Table 4-7. More interesting is the question whether Rank 1 and Rank 2 are different distributions. If so, this would strengthen the claim that we can guarantee a Top 1 ranking based on IoU scores. Table 4-7 shows that the confidence that Rank 1 and Rank 2 samples are different is very low. This strongly implies that the IoU score is not the defining value that can guarantee a Top 1 classification.



(a) Distributions per interval. **(b)** Distributions per rank.

Figure 4-11: The IoU scores, visualized in several boxplots showing the distribution of the scores per rank (interval). Here all samples that did not have a Top 1 score for the corresponding ground truth segmentation are omitted.

Table 4-7: Results of KS testing on different SACEI ranks versus the Top 1 ranking samples. The lower the p-value, the higher the certainty that the Top 1 samples and Top n samples are part of the same distribution.

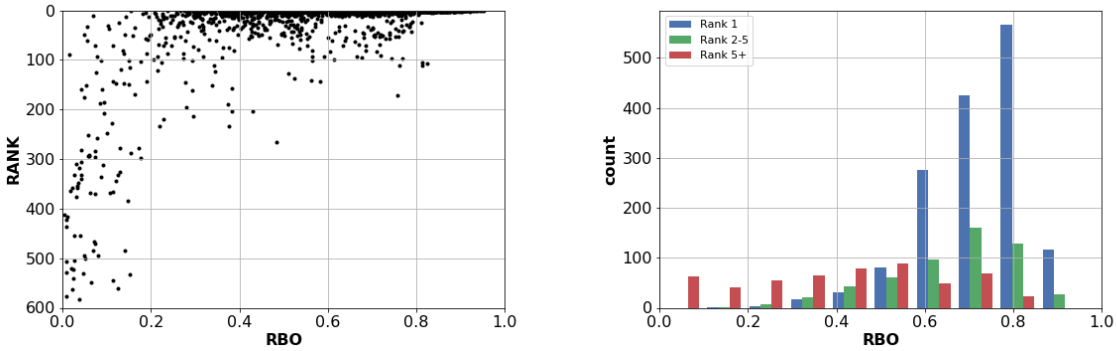
Rank	KS-statistic	p-value
Rank 2	0.078	3.866E-01
Rank 3	0.369	1.556E-06
Rank 4	0.252	7.506E-02
Rank 5	0.491	6.432E-03
Rank 2-5	0.166	3.543E-05
Rank 6-10	0.448	4.628E-10
Rank 10+	0.738	8.993E-62

4-3-2 RBO Scores

A drawback for evaluating performance using the SACEI rank is the need for the ship class ground truth. In the MWC Large set ship classes are present that are not incorporate into the SACEI database, and therefore cannot be classified correctly. We can however check the ranking returned when we present the classification algorithm with the ground truth segmentation, and subsequently compare that ranking to the ranking returned by the algorithm when presented with the network segmentation. Implementing the Rank-Biased Overlap (RBO) score (Section 3-3-3) as a measurement of rank similarity, we can investigate the relation between IoU and the output of the algorithm even further. A negative aspect of the evaluations on the MWC250 dataset was the lack of low scoring IoU samples. The segmentation networks are performing simply too well on the MWC250 set to return sufficient low IoU scores for evaluating the IoU versus classification score relationship. The MWC Large set presents a bigger challenge with the increasingly noise augmented levels. As depicted in Figure 4-6 and listed in Table 4-5, there are more poor segmentations with corresponding RBO scores that will be presented in this section.

MWC250

To evaluate the relationship between the RBO scores and the SACEI rank, we plot all samples from the MWC250 set in Figure 4-12a, showing the scatter relation between the RBO score and the corresponding SACEI rank. Two things are visible in this plot. There is a relation between a low RBO score and increasingly low SACEI ranking. This relation is far from linear, and is not very consistent, judged by the amount of outliers. Both low and high RBO scores can lead to a high SACEI rank. From all samples that have a RBO score of 0.8 or higher, 524 out of 529 rank in the top 5. In Figure 4-12b, we can clearly see that low ranking samples (Rank 5+) are randomly distributed on the interval $[0 - 0.8]$.

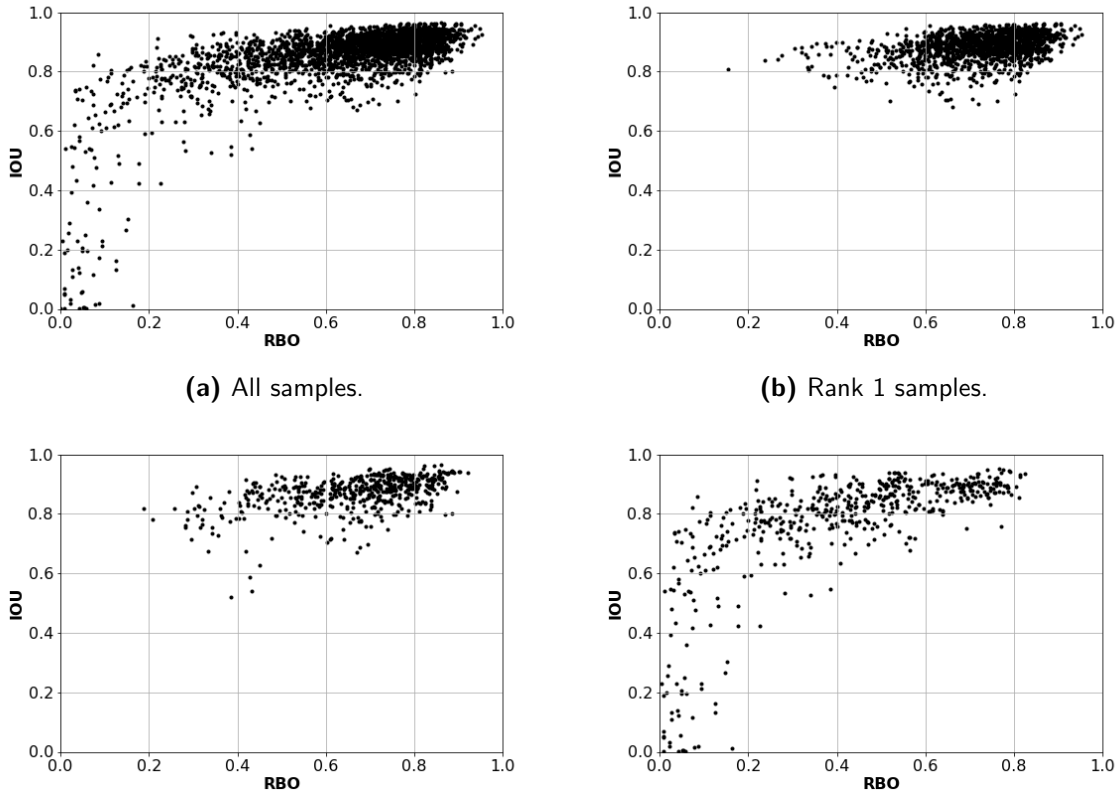


(a) The SACEI ranks plotted against the RBO score from the MWC250 set.

(b) Histogram showing how RBO values are distributed given a certain rank (interval).

Figure 4-12: Visualization of the relation between RBO and SACEI rank, includes samples from all networks listed in Table 4-6

To investigate the relation between RBO and IoU score, we repeat the procedure of plotting all values from the MWC250 dataset in a scatter plot. The result is shown in Figure 4-13. Due to the high performance of the evaluated networks on the MWC250 set, the amount of data points in Figure 4-13d is low. From Figure 4-13b we can see that there is indeed a difference in the clusters for top 1 and top 10+ ranking samples. Top 1 ranking samples have a higher mean RBO score, and a IoU score of less than 0.7 indicates a very low possibility to attain a classification on rank 1.



(a) All samples.

(b) Rank 1 samples.

(c) Rank 2-5 samples.

(d) Rank 5+ samples.

Figure 4-13: The IoU versus RBO results of each of the networks listed in Table 4-6. To show the location of result clusters of different SACEI ranks, multiple sub plots are shown with different SACEI ranks.

MWC Large

In Figure 4-14, the samples resulting from evaluation on the MWC Large set are shown. The samples with lower IoU scores, that were not present in the evaluation of the MWC250 set, complement the incomplete picture from Figure 4-13a. While there are more data points, the overall relation is similar in both the MWC250 set and the MWC Large set. However, the information we can draw from Figure 4-14 is limited. Without a ground truth class label, it is not possible to assess if the output of the SACEI classification algorithm is accurate.

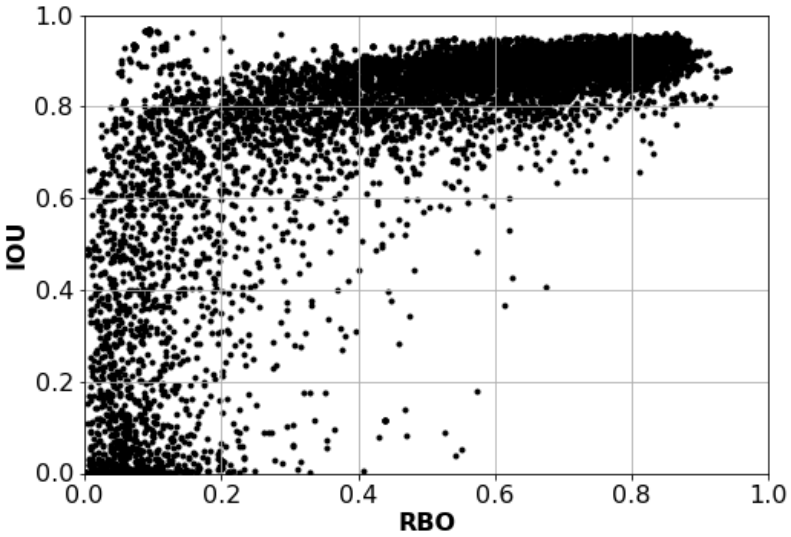


Figure 4-14: The SACEI ranks plotted against the RBO score from the MWC Large set, includes samples from all networks listed in Table 4-6.

Concluding on the RBO score

We can draw a few conclusions based on our previous evaluation of the relation between the RBO score and the resulting rank, as well as the relation between the IoU score and the resulting rank. Based on the results from the MWC250 set, we can conclude that there is a low probability that samples with a IoU score lower than 0.8 get a rank 1 score. This idea is reinforced in Figure 4-15, where we can also see that there is no distinction in distribution between the different rankings. It must be noted that there is a low amount of samples that are ranked 5+, giving us little information and confidence in the significance of the samples that score 0.7 IoU or less.

Similar to the IoU score, the RBO score has a lower boundary for confidence in high rankings as well. When the RBO score is higher than 0.6, the confidence in a top 5 score increases greatly. Based on Figure 4-13d and Figure 4-12b, we can assume that a RBO score of more than 0.8 almost guarantees a top 5 score. Due to this property, the RBO score is a more reliable rank estimator compared to the IoU score. This is not surprising, as the RBO score is derived from the classifier output, incorporating the non-linear effects of the classification algorithm.

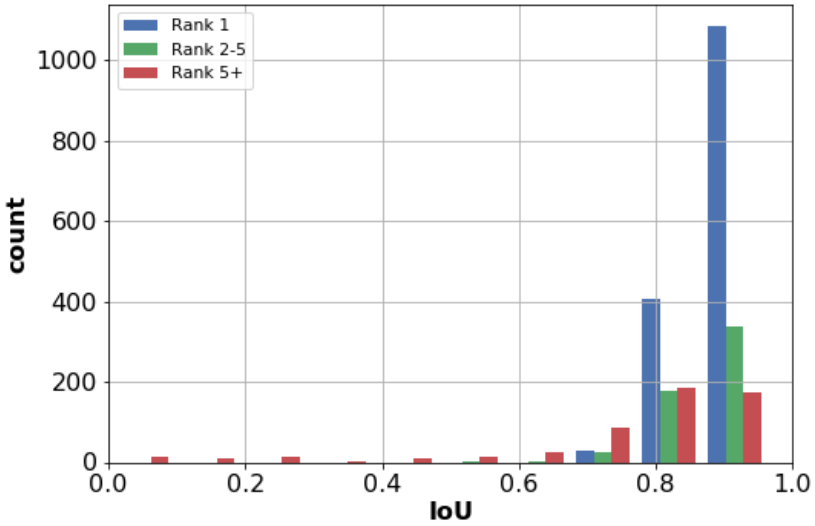


Figure 4-15: Histogram showing the distribution of IoU scores of different SACEI rank samples, evaluated on MWC250.

4-4 Ensemble Training

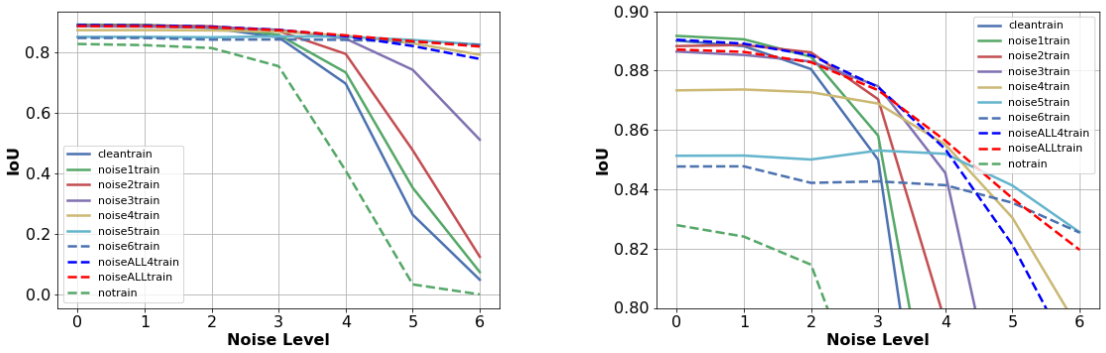
Research focusing on different noise levels in training data of deep learning networks has been carried out [40]. Different levels of noise were applied to a training set, and subsequently each distinct resulting network was evaluated on each and every noise level. In addition to this process, an ensemble data set containing all noise levels was considered. After evaluation, results show that the ensemble dataset outperformed the networks that were trained on a single noise level. The result is a robust network with higher performance on highly noisy images, and no trade-off on lower noise levels. This observation gives high motivation to evaluate the performance of ensemble datasets in the image segmentation domain.

4-4-1 Ensemble Data Setup

Creating an ensemble of different noise levels is not straightforward. Instead of sampling each parameter from the noise model from Section 3-2 separately, we choose to retain the noise levels shown in Table 3-2. This is done to research the influence of combining different dataset, rather than training on different noise parameters altogether. During the generation of the ensemble dataset, each sample is augmented with a uniformly sampled random noise level on the interval 0-6. From Table 4-5 we can see that training on noise level 5 and 6 only marginally improves evaluation results. Based on this observation, a second ensemble set is constructed, with noise levels sampled from the interval 0-4. The assumption is that this tighter interval will have less performance decrease on the low noise levels, while being less robust to high noise levels.

4-4-2 Results on MWC Large

We trained our DeepLab network on two additional noise augmented datasets based on MWC Large. This resulted in a robust network with overall high performance. The IoU scores, evaluated on different noise levels, are presented in Figure 4-16. Taking a closer look at the higher IoU levels (Figure 4-16b), we observe two things. We assumed that shifting the noise ensemble from all levels to only the interval $[0-4]$ would increase performance on the lower noise levels. This is indeed the case, compared to the full ensemble training. The trade-off that was an issue in the other networks, is considerably smaller when trained on an ensemble dataset. Based on the IoU evaluation on the MWC Large set, we conclude that ensemble training is a powerful technique for training a noise robust network, while retaining accuracy on clear images.



(a) Full view of the deteriorating effect of increasing noise levels on the evaluation set.

(b) A zoomed version, focused on the interval $[0.80-0.90]$.

Figure 4-16: IoU performance of all networks, including the networks trained on ensemble datasets.

4-4-3 Results on MWC250

Following up on the promising results on the MWC Large set, we move to evaluating the results from the SACEI classification algorithm. Before we evaluate the ranks, we revisit the MWC250 dataset in terms of IoU score. The results of the segmentations on the MWC250 set are depicted in Figure 4-17. Evaluation scores are not evidently higher, this can be confirmed in Table 4-8. While the ensemble networks are robust and perform well on all noise levels, MWC250 does not contain a uniformly sampled selection from those noise levels. To investigate whether the dataset diversity and sample size is the reason of an unexpected result, we test the significance of the difference between all distributions of IoU scores. This is depicted in Figure 4-19. From this figure we can derive that, given the results on the MWC250 set, no significant difference is measured between the best performing network, and the ensemble network. Based on the results from Figure 4-16, we expect that the effect of ensemble training would have been bigger if the MWC250 dataset contained more low quality images.

The resulting SACEI scores, including the scores of previously presented networks, are presented in Figure 4-18. The results show again the same inconsistency, where IoU scores

roughly indicate high ranks, but is no guarantee of a top 1 result. In Figure 4-18 we see that the top 5 performance of ensemble networks equals that of the other noise trained networks. This indicates that the lower top 1 score is a result of more top 2-5 ranking samples, compared to the single noise level trained networks. These inconsistencies give rise to the question, what are the differences in the segmentation result between high ranking and lower ranking SACEI results? We investigate this in Section 4-7.

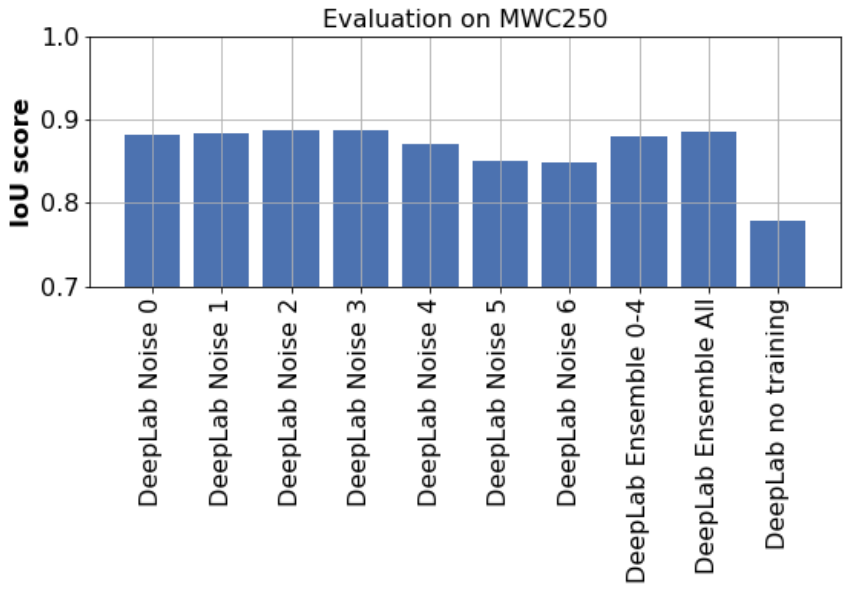


Figure 4-17: Mean IoU results from all trained DeepLab networks on the MWC250 set, including the ensemble trained networks.

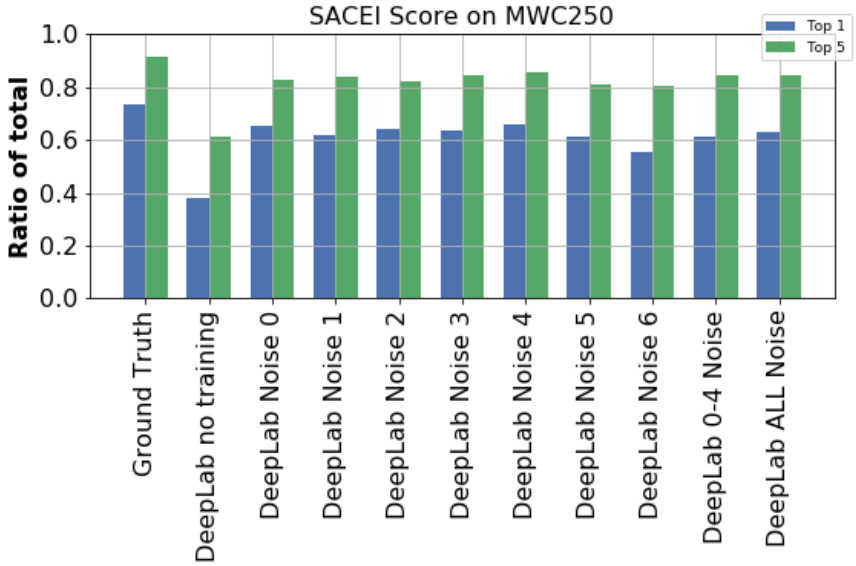


Figure 4-18: SACEI classification results on MWC250, both in Top 1 and Top 5 ratios.

Table 4-8: Results of all DeepLab networks on the MWC250 set, including the ensemble trained versions. Best result in bold.

Network	IoU	SACEI Top 1 Ratio	SACEI Top 5 Ratio
DeepLab no training	0.7788	0.378	0.610
DeepLab Noise 0	0.8809	0.656	0.830
DeepLab Noise 1	0.8842	0.618	0.838
DeepLab Noise 2	0.8866	0.639	0.822
DeepLab Noise 3	**0.8873**	0.635	0.846
DeepLab Noise 4	0.8712	**0.660**	**0.855**
DeepLab Noise 5	0.8505	0.610	0.809
DeepLab Noise 6	0.8487	0.552	0.805
DeepLab Ensemble 0-4	0.8806	0.610	0.846
DeepLab Ensemble All	0.8851	0.631	0.846

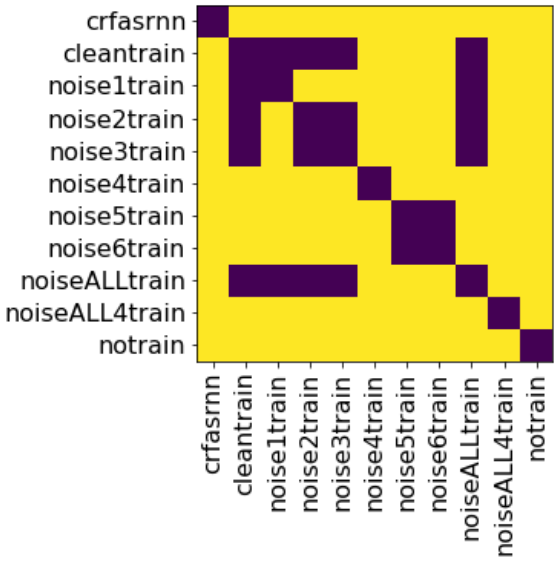


Figure 4-19: Result matrix showing the outcome of a Wilcoxon test between all results per network. Yellow shows a value of p<=0.01, indicating high certainty that two distributions are not similar. Purple shows distributions with p>0.01, indicating a low certainty that the two distributions are different.

4-5 Convergence of the Network

To assess if different MWC Large training sets affect the performance of the DeepLab network, we used the training parameters in Table 4-2. However, we did not evaluate whether this is an optimal setting for each and every dataset. An experiment is carried out, training the DeepLab network on the dataset comprised of noise levels 0-4. In this experiment we evaluate if decreasing or increasing the amount of training iterations affects the network IoU score.

The number of iterations used are 1000, 10.000, 25.000, 100.000 and 200.000. The results are shown in Figure 4-20.

The amount of iterations in training greatly influences the performance of the network, but this effect diminishes quickly after the 10.000 iterations used in other experiments. Nevertheless, increasing training time does have a positive effect, indicating that the network has not converged to an optimal point after 10.000 training iterations. While this is true for the Ensemble 0-4 data set, this is no guarantee that it holds for all other noise levels of the MWC Large data set. It does motivate to explore optimal hyper-parameter selection, in order to achieve better results with the same network and data set.

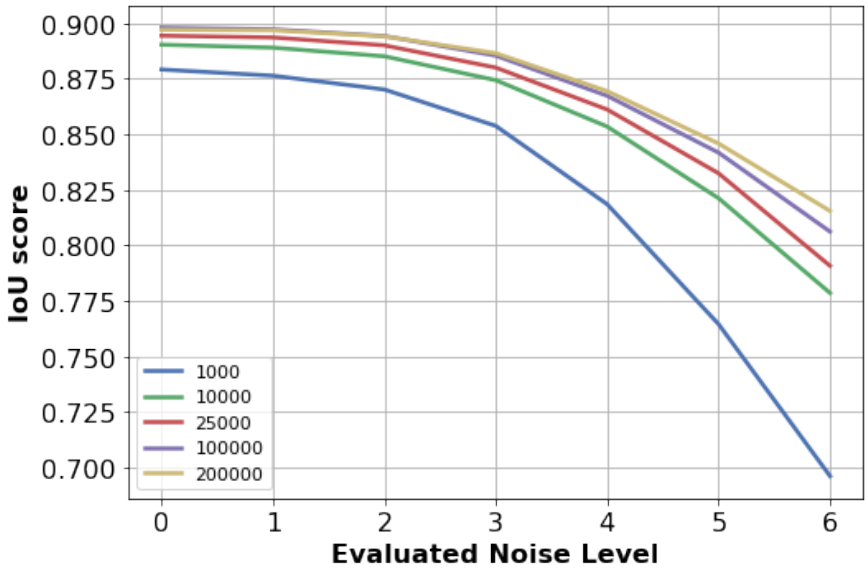


Figure 4-20: The IoU performance of DeepLab trained on the MWC Large Ensemble dataset, Noise Levels 0-4. To explore the influence of convergence, different amounts of training iterations are shown.

We also evaluate the newly trained networks on the MWC250 set, to investigate if this improvement on the MWC Large data set is also present on the SACEI ranking. The results are shown in Figure 4-21. We observe an increase in performance, beyond the performance of the previously best performing network (DeepLab Noise Level 4). With an increase in both IoU score over all noise levels on MWC Large, and best results on the SACEI score on MWC250, we conclude that the network hyper parameters remain an interesting field for further research.

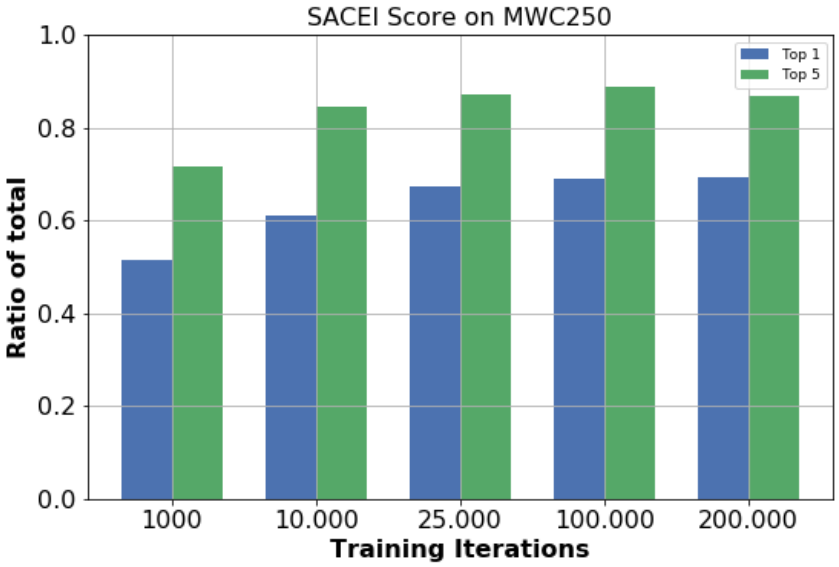


Figure 4-21: The SACEI performance of DeepLab trained on the MWC Large Ensemble dataset, Noise Levels 0-4. To explore the influence of convergence, different amounts of training iterations are shown.

4-6 Ensemble of Classifiers

A technique applied to different algorithms to boost performance is ensemble learning [41]. The goal is to combine the output of multiple classifiers, and improve the performance beyond the level of the best single classifier. In our case, there are not multiple different classifiers generating the segmentation maps, but networks trained on different noise levels. Here, we can consider differently trained network as distinct unique classifiers. This motivates us to investigate the results when combining the output of multiple networks.

In this section we implement a method to combine the SACEI ranks, returned by the classification algorithm for each separate network. To decide what output to use among all outputs from the single classifiers, we implement a majority vote [42]. The SACEI classifier returns a ranking, with all database ship Ids ranked from 1 to n, n being the amount of ships in the database. Per index of the returned ranking, we apply a majority vote along all networks. An example is shown in Table 4-9, indicating the structure of the outputs and the application of the majority vote algorithm.

The result of majority voting on the output of all DeepLab networks trained on a version of MWC Large is shown in Table 4-10. This includes the networks trained on the separate noise levels from 0 to 6, the ensemble datasets from Section 4-4, and the differently trained networks from Section 4-5. The performance is increased compared to the best network, DeepLab trained on Noise Level 4. This result motivates further research for the application of different ensemble of classifier approaches.

Table 4-9: An example of the majority vote applied to the SACEI output. Note that with an equal outcome (no majority), the lowest id number is returned. A second thing to note is when multiple occurrences of the same id are returned in the final majority voted output. Only the first occurrence should be considered, all duplicates can be discarded.

Index	Network 1	Network 2	Network 3	Majority Vote
1	23	23	42	23
2	42	42	41	42
3	132	41	23	23
⋮	⋮	⋮	⋮	⋮
n	243	142	534	142

Table 4-10: The counts of the SACEI results from a human operator (GT), the best performing network (DeepLab Noise 4), and a majority voted result over all DeepLab networks trained on a MWC Large set.

SACEI ranking	GT	DeepLab Noise 4	Majority Voted
Top 1	177	166	176
Top 5	220	221	221
Top 10	225	224	224

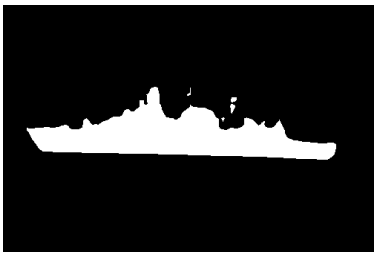
4-7 Visual Comparison of Segmentation Results

In the previous section we discovered that slight changes in IoU score can influence the SACEI rank. There is also a large range of IoU values that ultimately result in a high SACEI rank. We did not explicitly investigate the segmentation results during the procedure of analyzing the results in terms of IoU, SACEI rank, and RBO. In this section we visually evaluate segmentation results.

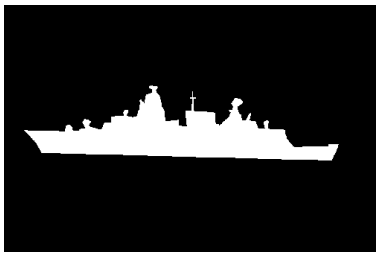
As mentioned in Section 3-3-2, some pixels are more important than others in the SACEI classification step. An example would be a pixel representing a structure on the ship. It adds to the uniqueness of the feature used for classification, the ship silhouette. An increasing number of missing structures in the segmentation will decrease the similarity between the segmentation and the correct database entry. A failure mode due to missing structures is shown in Figure 4-22. However, a pixel on the hull of the ship carries little to no information. They are even ignored, discarded during the conversion from segmentation to ship silhouette in the classification algorithm. A rank difference of 14 is observed in Figure 4-22, although the IoU score is not extremely low (<0.7).

In some cases two segmentations are very similar, yet still differ in rank. Figure 4-23 shows such a situation. With only minor differences in the segmentation, the resulting rank jumps from 1 to 35. Due to a gap in the mast structure, the resulting retrieved silhouette lacks this feature. Therefore, it is not representative for the actual vessel and ranks 35. In Figure 4-23 we see that a larger segmentation, containing more false positives, leads to the inclusion of small mast structures. Due to the higher amount of false positives, the IoU score drops, but the important mast structure feature is retained. Therefore, the resulting SACEI rank is higher.

(a) The original image.

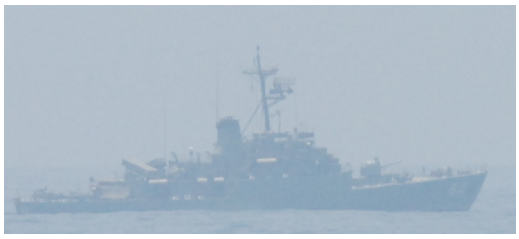


(b) A segmentation with a missing structure on top. The resulting SACEI rank is 15.



(c) The human operator annotation used as a ground truth for the IoU score. The resulting SACEI rank is 1.

Figure 4-22: A failure mode of a segmentation with an IoU score of 0.827. Only the biggest continuous part of the segmentation is used as a silhouette. The smaller, unconnected white parts are discarded during the extraction of the silhouette



(a) The original image.

(b) The human operator annotation used as a ground truth for the IoU score. The resulting SACEI rank is 2.

(c) A slightly different segmentation from a network trained on Noise Level 4. The resulting SACEI rank is 1.

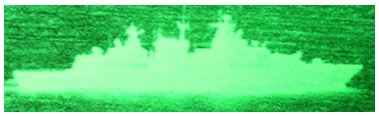
(d) A segmentation of a network trained on Noise Level 3, with a gap in a mast structure, which has only a small effect on the IoU score. The resulting SACEI rank is 35.

Figure 4-23: A failure mode of a segmentation with an IoU score of 0.862 (Figure 4-23d). The difference between the two segmentations in terms of IoU is 0.00483. Even with a low IoU difference, the SACEI rank difference is 34.

A different type of important pixels are the pixels that define the extreme points of the hull and stern of the ship. Due to a scaling operation in the classification step, missing or extra pixels will lead to a scaled version where not only the width of the ship is incorrect, but the structures on top are stretched or compressed when compared to the database. An example of this type of failure mode is shown in Figure 4-24.

A different case is focused on how even an inaccurate segmentation can lead to a top 1

classification from the SACEI algorithm. Not all ships are very closely related, based on their silhouettes. Some ships are so different from others, that even with a missing structure or inaccurate edges, a top 1 rank is returned. The opposite is occurring as well.



(a) The original image.

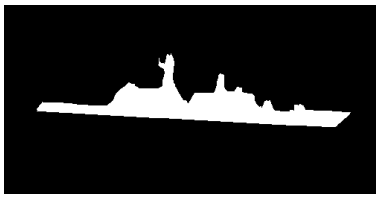
(b) A segmentation with a missing bow, causing low IoU and scaling issues. The resulting SACEI rank is 145.

(c) The human operator annotation used as a ground truth for the IoU score. The resulting SACEI rank is 1.

Figure 4-24: A failure mode of a segmentation with an IoU score of 0.668. Due to this low score and scaling issues resulting from the missing horizontal extremum pixels from the bow, a very low SACEI rank of 145 is returned.

(a) The original image.

(b) A segmentation with inaccurate structures, and overall inaccurate edges, causing a low IoU score. The resulting SACEI rank is 1.



(c) The human operator annotation used as a ground truth for the IoU score. The resulting SACEI rank is 1.

Figure 4-25: A mode of a segmentation with an IoU score of 0.786. Even with a low IoU score, a SACEI rank of 1 is returned.

Chapter 5

Conclusion

5-1 Conclusion on Research Questions

In Section 1-2, we presented two main research questions. Both contain multiple sub-questions, going deeper into relevant areas of research. The first question is focused on the selection of a network through a comparative study. The second question explores the possibilities for performance improvement, given a suitable network. In this section we will revisit each question, and present the conclusions drawn from the experiments in Chapter 4.

To select different networks, and subsequently assess the performance, we explored the different structures that are available in literature. We presented multiple solutions to counter the loss of spatial information during pooling operations. Networks employing these techniques are presented in Section 2-2. With this selection of networks (CRF-RNN, Mask R-CNN, PSPNet and DeepLabv3), we evaluated their performance on the MWC250 dataset. The evaluation was done both with the Intersection over Union (IoU) metric, and the SACEI classification algorithm. We observe that Mask R-CNN and DeepLab are the best performing networks on the IoU metric. However, Mask R-CNN does not recover edges with the quality required for classification. The results are also skewed due to the different detection failure rates, resulting in lower average IoU scores for the CRF-RNN network. The assumption that edges are important for classification accuracy was confirmed, as the CRF-RNN network outperforms Mask R-CNN on the SACEI classification task.

Based on this comparative study of multiple networks in Section 4-1, evaluated on different tasks and metrics, we conclude that the most promising network for our purposes is the DeepLab network. With a high score in the task of detection, high IoU scores, top performance on the SACEI classification task, and good edge reconstruction, it is the best choice among the considered networks.

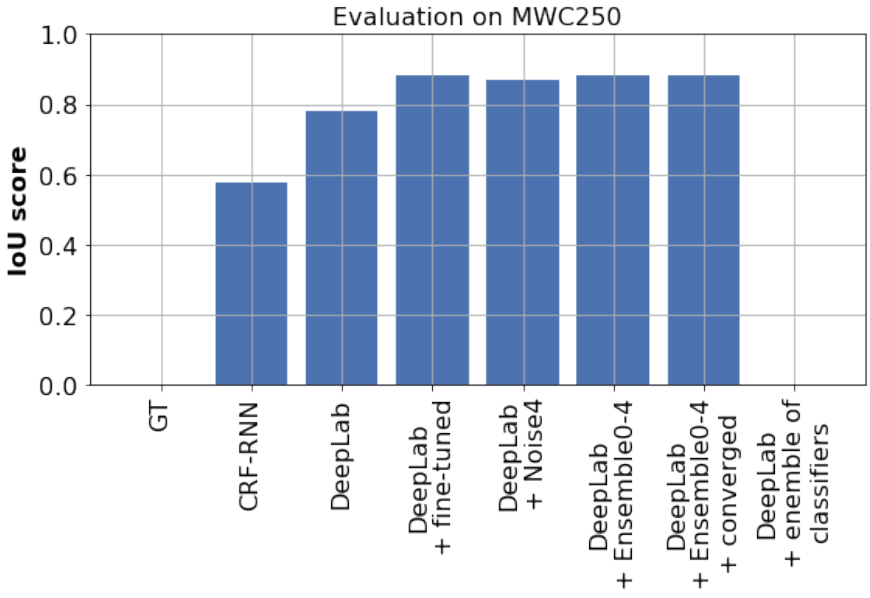
With a suitable candidate network for further experiments, we moved on to answering the second research question. Here we tried to solve the problems from Section 1-1, which resulted in increased overall performance. The first step was to assess whether training the network on data that is representative of the operational input images increases performance. For this

purpose, we used a dataset provided by the Maritime Warfare Centre (MWC) (MWC Large, Section 3-1-4). This fine-tuning procedure had a beneficial effect, as shown in Figure 5-1.

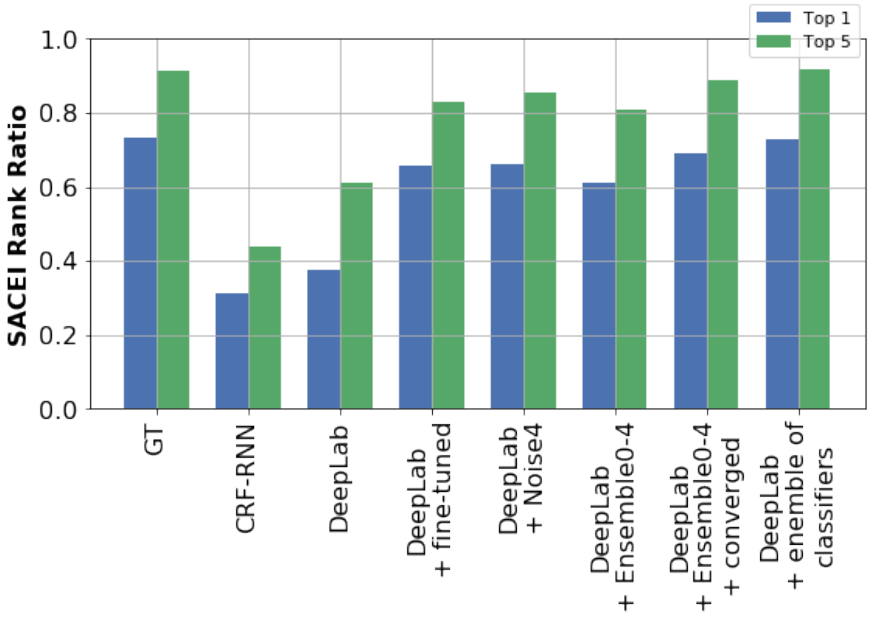
Another challenge in increasing performance, was developing robustness against noisy images. The degradation of the image quality, due to physical phenomena occurring during optical sensing at long distances over sea, has a negative impact on the performance of the segmentation networks. We developed a noise model (Section 3-2), imitating the disturbances present in operational input images. This model was used to augment our train data from the MWC Large dataset, and train DeepLab on the augmented data. Subsequently, noise augmented data was used to evaluate performance of differently trained DeepLab networks on several artificial noise levels. The evaluation results show a great increase in robustness to noise when trained on noisy data. A trade-off is present, between noise robustness and performance on clean images.

We combined the different noise levels to a single dataset, comprised of an ensemble of all noise levels. When training our DeepLab network on this dataset, we were able to increase the robustness of our network. By combining different noise levels, the trade-off we observed previously was greatly reduced. The result is a DeepLab network with high overall performance. The performance results are shown in Figure 4-16. To further increase performance, we explored the influence of different training lengths in Section 4-5. Applying a longer training length increased the performance on IoU score and SACEI rank. Additionally, we introduced an ensemble of classifiers in the form of a majority vote over the SACEI outputs (Section 4-6). This ensemble of classifiers further increased results beyond the best result achieved with a single network.

In Figure 5-1 we summarize the results achieved in this thesis. We show the original network selected before for use in the SACEI classification pipeline (CRF-RNN). Next to that, we show the best network from the comparative study in Section 4-1 (DeepLabv3). Finally, we present the improved results from the best networks resulting from noise augmented data training, an ensemble of noise augmented data training, and an ensemble of classifiers.



(a) IoU scores.



(b) SACEI ranking ratio.

Figure 5-1: Results in terms of IoU and SACEI rank ratios on the MWC250 test dataset. Depicted is the performance of CRF-RNN, which is the network currently used in SACEI for automatic segmentation. Additionally, the results achieved by the work in this thesis with the DeepLab network are shown for comparison. Note that ensemble of classifiers is not applicable to the IoU metric, and is therefore omitted from Figure 5-1a.

5-2 Recommendations

In this thesis efforts were made to optimize the performance of an automated approach for segmenting digital ship images, with regards to the final classification result. A comparative study in Section 4-1 shows that DeepLabv3 is a suitable candidate for this task, based on its high detection rate and IoU performance on the MWC250 set. Subsequently, performance of the DeepLab network was improved by training on a larger dataset comprised of high quality ship images (MWC Large). Robustness was increased by training on different configurations of noise augmentation on the training images.

Based on the high overall IoU scores of the DeepLab network trained on an ensemble of noise levels 0-4, this configuration is recommended for implementation in a single network setting. This configuration does not achieve the best results in terms of SACEI score, but it is expected that operational images are of lesser quality compared to the images in the MWC250 set. In that situation, the network will benefit more of its robustness compared to the application on the MWC250 set. Based on the results in Section 4-6, we suggest to extend the single network classification to an ensemble of classifiers. This approach has proven to increase performance compared to a single network approach. The best method for an ensemble of classifiers should be further researched, but a majority vote over the results of DeepLab networks in Section 4-4 already achieved best results, motivating additional research in this field.

In light of the automation aspect of the classification pipeline, it is suggested to apply the DeepLab network as a Region of Interest (ROI) detector. This will further eliminate the need for a human operator in the process. A first pass of the input image can be done with the DeepLab network. The resulting segmentation map can be used to detect a ROI. Subsequently, the original image can be cropped to the detected ROI, and passed to the network for a second segmentation. Finally, the second segmentation map is used for classification.

Additionally, an effort was made to assess the predictive capacity of the IoU and Rank-Biased Overlap (RBO) metrics. While for both scores a high value indicates a high ranking classification result, the certainty of the prediction is very low with a high amount of outliers and inconsistent behavior. It is therefore recommended to assess network classification results on quality by comparing the SACEI classification output to a ship class ground truth.

In Section 4-5 we experimented with different training lengths. We showed that there is room for improvement in terms of performance, through experimenting with different network parameters. Further research is suggested in the field of the optimization algorithm used, because an optimally tuned Stochastic Gradient Descent (SGD) approach can outperform other optimizers [21]. Additionally, DeepLab has other parameters not fully explored in this thesis that might well increase performance, if optimally tuned.

A problem encountered when comparing the SACEI results from different networks were the inconsistent results with respect to their respective IoU score on MWC250. A data set, with ship class ground truths, that has more variety of images in terms of quality can give a better representation of the performance during operation. Constructing a test set of operational images would be a useful tool in determining if a network can be moved to operation from development.

In Section 4-6 we explored the influence of applying an ensemble of classifiers. The decision to select a majority vote in this situation was made based on the ease of implementation. The

approach has a drawback in the sense that duplicate Ids in the returned ranking are possible, and therefore Ids can be omitted too. A study comparing different approaches to an ensemble of classifiers is recommended, eliminating some drawbacks and improving results.

A fundamental problem in the optimization of a deep learning network is the choice of a loss function. In most segmentation cases the cross-entropy loss is used to drive the optimization, but in our case this objective is disjunct with our final objective, a high SACEI rank. This is highlighted in Section 2-1-7. It is recommended to explore the possibilities of using a loss function that better represents the final objective, or at least experiment with differently weighted pixels in an image based on their importance during classification. The failure modes presented in Section 4-7, with emphasis on the important structures, could be eliminated with such an approach.

We have shown that with an open source network trained on multiple classes, a project specific task (such as a ship/background only segmentation) can be performed. More interestingly, we used a relatively small data set of only 2500 training images, making it a feasible task to construct such a dataset for a different application. The images in this dataset were mostly of ships in good conditions. Through data augmentation, we trained the network to not only perform on good conditions, but also on poor conditions that we simulated in the training set. Our results show that it is possible to fine-tune a network such as DeepLab effectively on a class specific task, with higher accuracy compared to the original network. This motivates to use this framework in other situations where the interest lies in segmenting specific classes with only a small dataset, especially when robustness to noise is a design objective.

Bibliography

[1] S. Zheng et al. "Conditional random fields as recurrent neural networks". In: *Proceedings of the IEEE International Conference on Computer Vision* 2015 Inter (2015), pp. 1529–1537. ISSN: 15505499. arXiv: arXiv:1502.03240v1.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *NIPS'12 Proceedings of the 25th International Conference on Neural Information Processing Systems* (2012), pp. 1097–1105. ISSN: 00010782.

[3] Y. Le Cun et al. "Handwritten digit recognition: applications of neural network chips and automatic learning". In: *IEEE Communications Magazine* 27.11 (1989), pp. 41–46. ISSN: 0163-6804.

[4] Jia Deng et al. "ImageNet: A large-scale hierarchical image database". In: *2009 IEEE Conference on Computer Vision and Pattern Recognition* (2009), pp. 248–255. ISSN: 1063-6919.

[5] O. Russakovsky et al. "ImageNet Large Scale Visual Recognition Challenge". In: *International Journal of Computer Vision* 115.3 (2015), pp. 211–252. ISSN: 15731405. arXiv: 1409.0575.

[6] D. G. Lowe. "Distinctive image features from scale-invariant keypoints". In: *International Journal of Computer Vision* 60.2 (2004), pp. 91–110. ISSN: 09205691. arXiv: 0112017 [cs].

[7] B. E. Boser, I. M. Guyon, and V. N. Vapnik. "A Training Algorithm for Optimal Margin Classiiers". In: *Proceedings of the fifth annual workshop on Computational learning theory* (1992), pp. 144–152. ISSN: 0-89791-497-X. arXiv: arXiv:1011.1669v3.

[8] J. Long, E. Shelhamer, and T. Darrell. *Fully Convolutional Networks for Semantic Segmentation*. 2014. arXiv: 1411.4038.

[9] Y. Le Cun et al. "Handwritten Digit Recognition with a Back-Propagation Network". In: *Advances in Neural Information Processing Systems* (1990), pp. 396–404. ISSN: 1524-4725. arXiv: 1004.3732.

[10] K. Fukushima. "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position". In: *Biological Cybernetics* 36.4 (1980), pp. 193–202. ISSN: 03401200. arXiv: arXiv:1011.1669v3.

[11] Y. LeCun, K. Kavukcuoglu, and C. Farabet. "Convolutional networks and applications in vision". In: *ISCAS 2010 - 2010 IEEE International Symposium on Circuits and Systems: Nano-Bio Circuit Fabrics and Systems* (2010), pp. 253–256. ISSN: 02714302.

[12] L.-C. Chen et al. *Rethinking Atrous Convolution for Semantic Image Segmentation*. 2017. arXiv: 1706.05587.

[13] Y. LeCun, Y. Bengio, and G. Hinton. "Deep learning". In: *Nature* 521.7553 (2015), pp. 436–444. ISSN: 0028-0836.

[14] C. M. Bishop. *Pattern Recognition and Machine Learning*. 2006. ISBN: 978-0-387-31073-2. arXiv: arXiv:1011.1669v3.

[15] O. Ronneberger, P. Fischer, and T. Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 9351 (2015), pp. 234–241. arXiv: 1505.04597.

[16] V. Badrinarayanan, A. Kendall, and R. Cipolla. *SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation*. 2015. arXiv: 1511.00561.

[17] K. Murphy. *Machine Learning: A Probabilistic Perspective*. 1991, pp. 73–78,216–244. ISBN: 9780262018029. arXiv: 0-387-31073-8.

[18] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016.

[19] H. Robbins and S. Monro. "A stochastic approximation method". In: *The annals of mathematical statistics* (1951), pp. 400–407.

[20] D. P. Kingma and J. Ba. "Adam: A Method for Stochastic Optimization". In: (2014), pp. 1–15. ISSN: 09252312. arXiv: 1412.6980.

[21] A. C. Wilson et al. "The Marginal Value of Adaptive Gradient Methods in Machine Learning". In: Nips (2017). ISSN: 10495258. arXiv: 1705.08292.

[22] L.-C. Chen et al. *DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs*. 2016. arXiv: 1606.00915.

[23] K. Simonyan and A. Zisserman. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. 2014. arXiv: 1409.1556.

[24] Xuming He, R. Zemel, and M. Carreira-Perpinan. "Multiscale conditional random fields for image labeling". In: *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.* 2 (2004), pp. 695–702. ISSN: 1063-6919.

[25] P. Krähenbühl and V. Koltun. "Parameter Learning and Convergent Inference for Dense Random Fields". In: *Proceedings of the 30th International Conference on Machine Learning (ICML-13)* 28 (2013), pp. 513–521. ISSN: 1938-7228.

[26] P. Krähenbühl and V. Koltun. *Efficient Inference in Fully Connected CRFs with Gaussian Edge Potentials*. 2012. arXiv: 1210.5644.

[27] K. He et al. *Deep Residual Learning for Image Recognition*. 2015. arXiv: 1512.03385.

[28] K. He et al. "Mask R-CNN". In: (2017). arXiv: 1703.06870.

[29] H. Zhao et al. *Pyramid Scene Parsing Network*. 2016. arXiv: 1612.01105.

[30] R. Girshick. "Fast R-CNN". In: *Proceedings of the IEEE International Conference on Computer Vision* 2015 Inter (2015), pp. 1440–1448. ISSN: 15505499. arXiv: 1504.08083.

[31] R. Girshick et al. "Rich feature hierarchies for accurate object detection and semantic segmentation". In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2014), pp. 580–587. ISSN: 10636919. arXiv: 1311.2524.

[32] M. Everingham et al. "The pascal visual object classes (VOC) challenge". In: *International Journal of Computer Vision* 88.2 (2010), pp. 303–338. ISSN: 09205691.

[33] T. Y. Lin et al. "Microsoft COCO: Common objects in context". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 8693 LNCS.PART 5 (2014), pp. 740–755. ISSN: 16113349. arXiv: 1405.0312.

[34] J. Tremblay et al. "Training Deep Networks with Synthetic Data: Bridging the Reality Gap by Domain Randomization". In: (2018). arXiv: 1804.06516.

[35] H.-C. Lee. "Review of image-blur models in a photographic system using the principles of optics". In: *Optical Engineering* 29.5 (1990), p. 405. ISSN: 00913286.

[36] R. B. Inampudi, T. N. Purimetla, and P. G. Satyanarayana. "Contrast degradation for improving quality of an image". In: *Geoscience and Remote Sensing Symposium, 2002. IGARSS '02. 2002 IEEE International* 6.C (2002), pp. 3408–3410.

[37] J. Nakamura. *Image Sensors and Signal Processing for Digital Still Cameras*. 2006. ISBN: 978-0-8493-3545-7.

[38] W. Webber, A. Moffat, and J. Zobel. "A similarity measure for indefinite rankings". In: *ACM Transactions on Information Systems* 28.4 (2010), pp. 1–38. ISSN: 10468188.

[39] M. Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. 2015.

[40] R. P. J. Nieuwenhuizen, D. D. Bouwman, and K. Schutte. "Assessing the prospects for robust sub-diffraction limited super-resolution imaging with deep neural networks". In: 2018.

[41] M. Thoma. *A Survey of Semantic Segmentation*. Tech. rep. 2016, pp. 1–16. arXiv: 1602.06541.

[42] M. Zhu. *When is the majority-vote classifier beneficial?* Tech. rep. 2013, pp. 1–17. arXiv: 1307.6522.

Glossary

List of Acronyms

ANN	Artificial Neural Network
CNN	Convolutional Neural Network
FCN	Fully Convolutional Neural Network
CRF	Conditional Random Field
MWC	Maritime Warfare Centre
SACEI	System for Automatic Classification of EO/IR/ISAR imagery
EO	Electro-Optical
LSVRC	Large Scale Visual Recognition Challenge
VOC	Pascal Visual Object Classes
MSCOCO	Microsoft Common Objects in Context
RGB	Red Green Blue
RBO	Rank-Biased Overlap
IoU	Intersection over Union
ROI	Region of Interest
KS	Kolmogorov-Smirnov
ASPP	Atrous Spatial Pyramid Pooling
SGD	Stochastic Gradient Descent