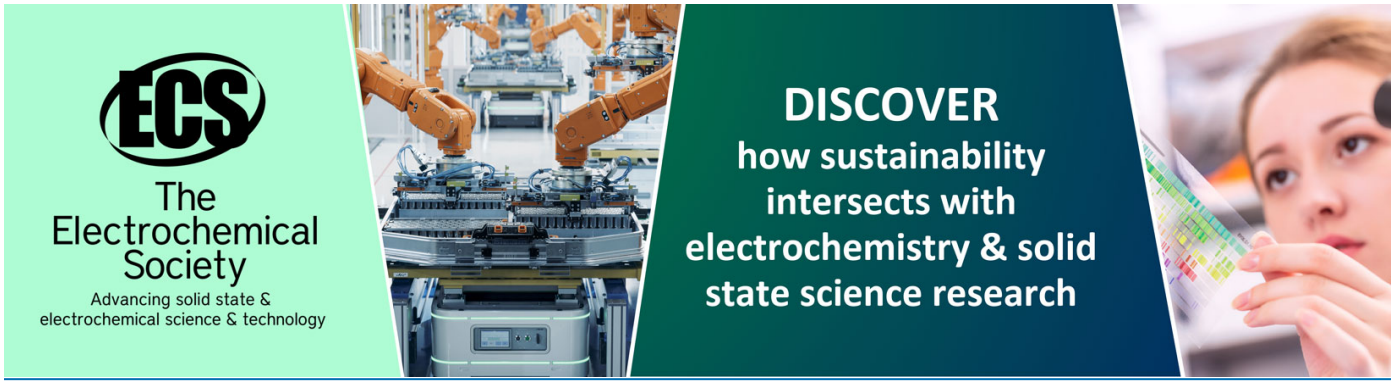
Lipschitz Constrained Neural Networks for Robust Object Detection at Sea



IOP Conf. Series: Materials Science and Engineering **929** (2020) 012023 doi:10.1088/1757-899X/929/1/012023

Lipschitz Constrained Neural Networks for Robust Object Detection at Sea

Jonathan Becktor, Frederik Schöller, Evangelos Boukas, Mogens Blanke, Lazaros Nalpantidis

Department of Electrical Engineering, DTU - Technical University of Denmark, Denmark

E-mail: jbibe@elektro.dtu.dk

Abstract. Autonomous ships rely on sensory data to perceive objects of interest in their environment. Deep Learning based object detection in the image domain commonly used to solve this issue. The robustness of such approaches in non-ideal conditions is, however, still to be proven. In this work state of the art methods are applied on the RetinaNet architecture attempting to create a more robust object detection network given noisy input data. The GroupSort activation function and Spectral Normalization is used and the results are compared to the standard RetinaNet network. Our findings show that these modifications perform better and ensure robustness under moderate noise levels, than the standard RetinaNet network.

1. Introduction

Traversing the ocean can present many challenges to an autonomous vessel. A diverse set of weather conditions, equipment malfunctioning, continuous movement all need to be taken into account. The task of classifying boats and buoys at sea, as depicted in figure 1, might work reliably under normal conditions [13] harnessing the descriptive power of neural networks [12]. However, uncontrolled changes of the input—as they are to be expected in an autonomous marine system–can result in radically different outcomes. Some of the inherent weaknesses of neural networks come from their internal workings. A neural network is a data driven, directed non cyclical graph where each level in the graph is a layer. Each graph node has a weight and bias followed by a nonlinear activation function, such as a rectified linear unit (ReLU), hyperbolic tangent, or sigmoid function. Weights and biases are updated with the gradient of

Figure 1. Example image (cropped for better visibility) of annotated data used in this paper.

a distance function, given the network output and the correct target. Thus, given a neural network, very slight changes to the input can have compounding effects on each layer and result in very different output. Exposing the pipeline to noisy input, such as rain, snow, adversarial noise, and varying lighting conditions, can all have detrimental effects on the outcome of neural networks [14] and computer vision systems in general [11].

The *goal* of this paper is to explore architectures that add robustness to object detection against noisy input in marine environments. Lipschitz constrained neural networks [4] have been shown to alleviate a lot of these issues [1]. As a result, in this work, we apply Lipschitz constrained architectures to confirm their applicability and robustness in the autonomous marine domain.

2. Related Work

This work focuses on the use of deep neural networks to achieve robust object detection at sea. Previous approaches has been along the lines of [5] where detection of small surface vessels was achieved by using Scale Invariant Feature Transform (SIFT) in concert with bag-of-features. In [13], a comparison of using RGB, long wave infrared (LWIR), and near infrared (NIR) data within the Resnet-50 RetinaNet [9] architecture is provided, followed by a thorough evaluation of the networks. The RetinaNet architecture, as introduced in [9], is a one stage detector. This means that an input only needs to be inferred once due to a novel loss function, thus increasing the speed. The need for robustness of neural networks is addressed in [14], where the authors constrain the Lipschitz constant of a network to be lower than 1 by constraining each layer's weight matrices to be orthonormal. Furthermore, [1] explains how training neural networks under a strict Lipschitz constraint is useful for provable adversarial robustness, generalization bounds, and interpretable gradients. The paper, introduces a Lipschitz constrained activation function and the Björck convolution, a convolution that ensures layer-wise orthonormality and thereby being 1-Lipschitz. Spectral Norm introduced in [10] introduces another tool to maintain the Lipschitz continuity. The paper describes a layer-wise regularization step that normalizes the largest singular value of each weight matrix to be less than one. In the works of [4], a method of constraining the Lipschitz constant is explored, the authors apply an orothonormality constraint on all weights of the neural network and can thereby regularize the total network to be 1-Lipschitz.

3. Background

In this section, the possible architecture modifications will be presented. This analysis will constitute the base for defining the proposed approach in the following section.

Lipschitz continuity. Given two real values x and y a function is Lipschitz continuous if and only if the norm of the functions subtracted is less or equal norm of the inputs multiplied by some λ see figure 2.

$$|f(x) - f(y)| \leq \lambda |x - y| \tag{1}$$

λ is the Lipschitz constant, which is also what is constrained in a Lipschitz constrained network. For a network to be Lipschitz continuous it suffices to compose 1-Lipschitz layers and activations.

GroupSort. GroupSort, as introduced in [1], is a Lipschitz continuous activation function, it sorts the pre-activated weights into some selected grouping. For example a GroupSort with a grouping size of 2, will split the weights into groups of 2, sort the weights in the groups, as can be seen done in figure 3. This operation is a nonlinear operation which is differentiable as the Jacobian is a permutation matrix which also preserve every p-norm [1]. Selecting a group of 2

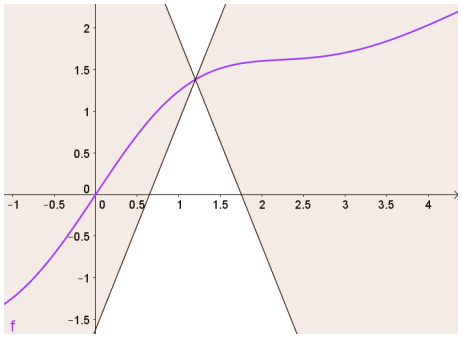


Figure 2. Lipschitz continuous function. Figure from [1]

is equivalent to the Orthogonal Permutation Linear Unit [3] also called MaxMin [1]. Whereas only using 1 group, e.g. sorting the entire slice, the operation is called FullSort.

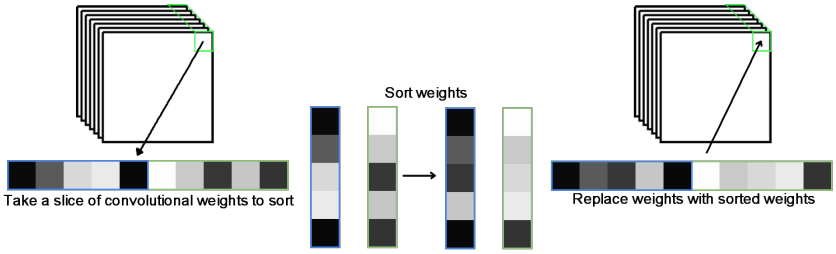


Figure 3. GroupSort with grouping size of 2, weights are sliced, sorted and replaced.

Spectral Norm. Spectral norm is a layer-wise normalization scheme introduced in [10] that enforces the largest singular value of the weight matrix to be less than 1 at each layer. This is achieved by estimating the singular vectors using power iteration and normalizing with respect to the found singular values:

$$\sigma(W) := \max_{h:h\neq 0} \frac{\|Wh\|_2}{\|h\|_2} = \max_{\|h\|_2\leq 1} \|Wh\|_2 \tag{2}$$

Where W are the weights and h is the largest singular value found by power iteration.

Björck convolution. The Björck convolution [2] is an iterative layer that lets us approximate a orthonormal weight matrix which ensures that singular values are exactly 1, which is achieved by iteratively applying the Taylor expansion of the polar decomposition.

Parseval Network. The Parseval regularization requires orthogonal weight matrices, this can be achieved with optimizing on the Stiefel manifold which is too expensive and can therefore be approximated with the layer-wise regularizer to ensure *Parseval Tightness* given by:

$$R_\beta(W) = \frac{\beta}{2} \left\| W^T W - I \right\|_2^2 \tag{3}$$

[1] `https://en.wikipedia.org/wiki/Lipschitz_continuity#/media/File:Lipschitz_Visualisierung.gif`

Optimizing $R_\beta(W)$ to convergence is too expensive, therefore the authors recommend only taking one step of descent, where the gradient is used to update the weight matrices by $W \leftarrow W(1+\beta) - \beta WW^TW$ Where β is the gradient step size.

4. Proposed Approach

The base architecture used is RetinaNet [9] with a ResNet-50 [6], Feature Pyramid Network [8] (FPN) backbone, as shown in figure 4. As discussed earlier, such networks can suffer from instability when exposed to noisy input. To combat the instability of the chosen neural networks, the architecture is modified to be 1-Lipschitz, as explored in [4] and [1]. As discussed above, the work of [4] proposes a regularization of the weights applied to each training loop given orthonormal weight matrices for each layer. On the other hand, [1] introduces the activation GroupSort (a sorting activation) and the Björck [2] convolution.

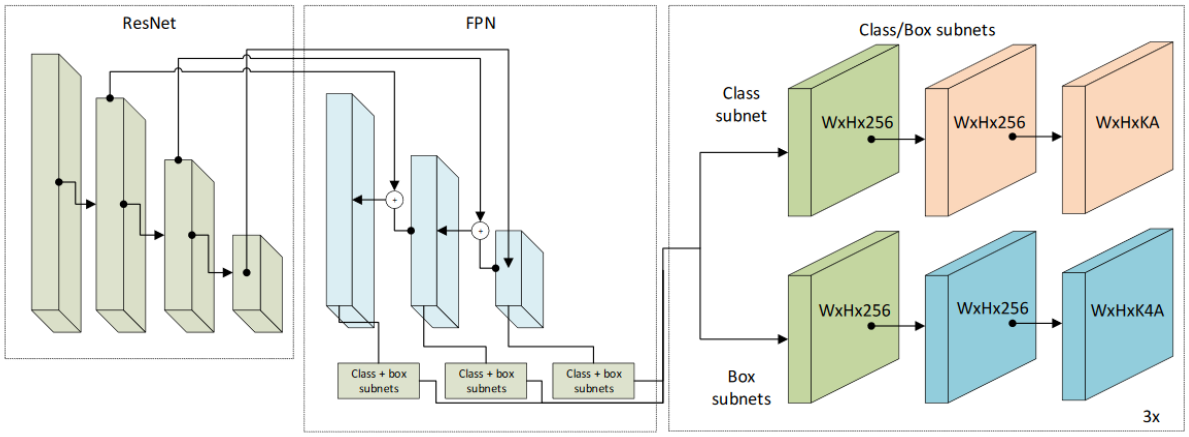


Figure 4. RetinaNet architecture.

The work done in [1] identified GroupSort activation and the Björck Convolution to be the most promising approach. However, using the Björck convolution on the full RetinaNet in its current form, is not feasible. Björck convolution can be applied to the classification/regression networks and use Spectral Normalization on the ResNet base.

Spectral Normalization is used as the regularization of the convolutions. Thus the selected modifications are GroupSort and spectral Normalization to ensure the Lipschitz constraint and thereby improving network robustness. These are selected as they are the least computationally expensive and the fastest to implement. The architectural modifications will be evaluated using similar methods as described by [13]. Several networks will be trained with differing GroupSort group sizes and will be compared to an equivalent network without any modification.

5. Experimental Setup

5.1. Dataset

The effectiveness of the selected networks architectures are evaluated on the dataset, consisting of a total of 9229 images with 20150 annotations divided into two classes namely; boat (6256) and buoy (13894). Each image was captured as a 2560×2048 pixels RGB image and subsequently downsized to 1440×1080 pixels. Data was acquired on-board ferries operating in the Southern Funen Archipelago. Images were captured every second from midday till dusk, ensuring a broad range of scene illumination(s). This yielded a training set containing 8306 images and a validation set containing 923 images. The image in figure 1 shows a cropped section of an annotated image from the dataset.

5.2. Training of the Neural Networks

The RetinaNet is trained with the ADAM optimizer [7], with an initial learning rate of 10^{-5} with a plateau scheduler. Each model is trained for approximately 110 epoch using early stopping, with a batch size of 2. There is a big difference in training time as the operations for 1-Lipschitz networks are computationally demanding.

6. Experimental Evaluation

The mean Average Precision metric (mAP) will be used as the method for comparing the selected networks. Average Precision is the area under the Precision-Recall curve shown in figure 5. Precision is found by $\frac{TP}{FP+TP}$ and recall with $\frac{TP}{FN+TP}$ where TP is true positive, FP is false positive and FN is false negative. Precision and Recall both change with the certainty threshold set for the network, which in these experiments is set to 0.5; this threshold can range from 0 (where everything is accepted) to 1 (where the network has to be extremely certain). The mAP is then found by the mean of the AP of each class in the system.

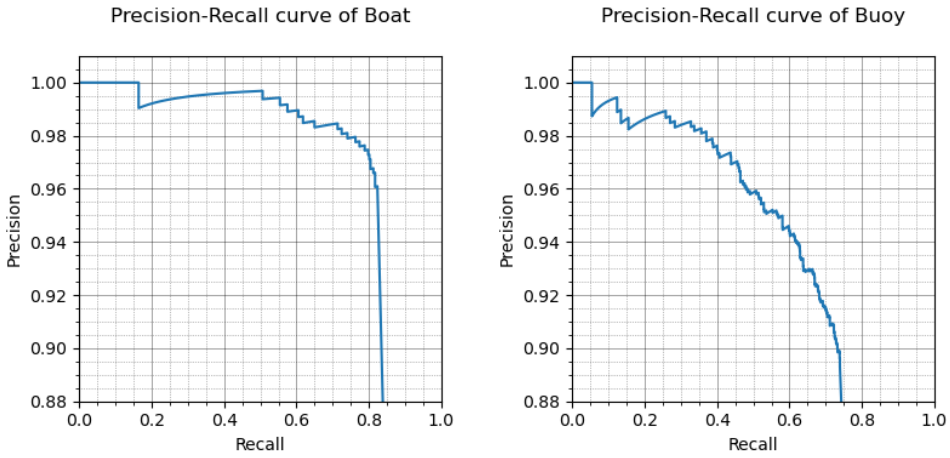


Figure 5. Precision-Recall curves for Boat and Buoy.

The selected networks will be tested on 3 different image corruption techniques; Gaussian noise, Gaussian Blur, and Salt and Pepper noise with increasing levels of intensity. The techniques were chosen as they loosely resemble the outcome of certain problems that could be expected during long-term operation of imaging sensors in harsh marine environments. Furthermore, to gather a deeper insight the precision and recall with respect to the area of each bounding box will be explored; this will provide insight on the size of objects the network has difficulty detecting.

6.1. Gaussian Noise

Corrupting an image with Gaussian noise is achieved by varying the intensities values of each color channel (R,G,B) for each pixel by some value drawn from a normal distribution. An example image, corrupted with various levels of Gaussian noise is shown in figure 6.

Table 1 summarizes the mean Average Precision (mAP) results for various levels of corruption of the images using Gaussian noise, in the following tables, GS is GroupSort and SN is Spectral Normalization. For each RetinaNet model the activation function is indicated, where ReLU corresponds to the implementation of [13], while the 3 remaining entries are our considerations.

Finally, figure 7 and figure 8 show the detection histograms, precision and recall as a function of the bounding box area for boats and buoys respectively. These figures compare the standard

Figure 6. Images corrupted with different levels of Gaussian noise. From left to right: no noise, $\mathcal{N}(0, 0.05)$, $\mathcal{N}(0, 0.075)$.

Table 1. mAP Results for various levels of Gaussian Noise. All results are ± 0.02

RetinaNet Model	No Noise	$\mathcal{N}(0, 0.025)$	$\mathcal{N}(0, 0.05)$	$\mathcal{N}(0, 0.075)$
ReLU	0.76	0.65	0.47	0.32
MaxMin, SN	0.75	0.68	0.49	0.29
GS 4, SN	0.76	**0.71**	**0.54**	0.32
GS 8, SN	**0.77**	0.70	0.50	**0.35**

ReLU network (figures on the left) versus the proposed "GroupSort 8, SN" network (figures on the right) for increasing levels of induced Gaussian noise (increasing from top to bottom).

6.2. Gaussian Blur

Corrupting an image with Gaussian blur is achieved by convolving the input images with a 2D Gaussian kernel. Increasing standard deviation of the used kernel results in a stronger blurring effect. An example image, corrupted with various levels of Gaussian noise is shown in figure 9.

Table 2 summarizes the mean Average Precision (mAP) results for various levels of corruption of the images using Gaussian Blur.

Table 2. mAP Results for various levels of Gaussian Blur. All results are ± 0.02

RetinaNet Model	No Noise	$\sigma = 1$	$\sigma = 1.5$	$\sigma = 2$
ReLU	0.76	0.69	0.66	**0.58**
MaxMin SN	0.75	0.71	0.66	0.52
GS 4, SN	0.76	0.73	0.69	0.56
GS 8, SN	**0.77**	**0.74**	**0.700**	0.55

6.3. Salt and Pepper Noise

Corrupting an image with Salt and Pepper noise is achieved by randomly selecting a percentage of pixels and assigning the minimum or maximum intensity value to them. An example image, corrupted with various percentages of pixels affected by Salt and Pepper noise is shown in figure 10.

Table 3 summarizes the mean Average Precision (mAP) results for various levels of corruption of the images using Salt and Pepper noise.

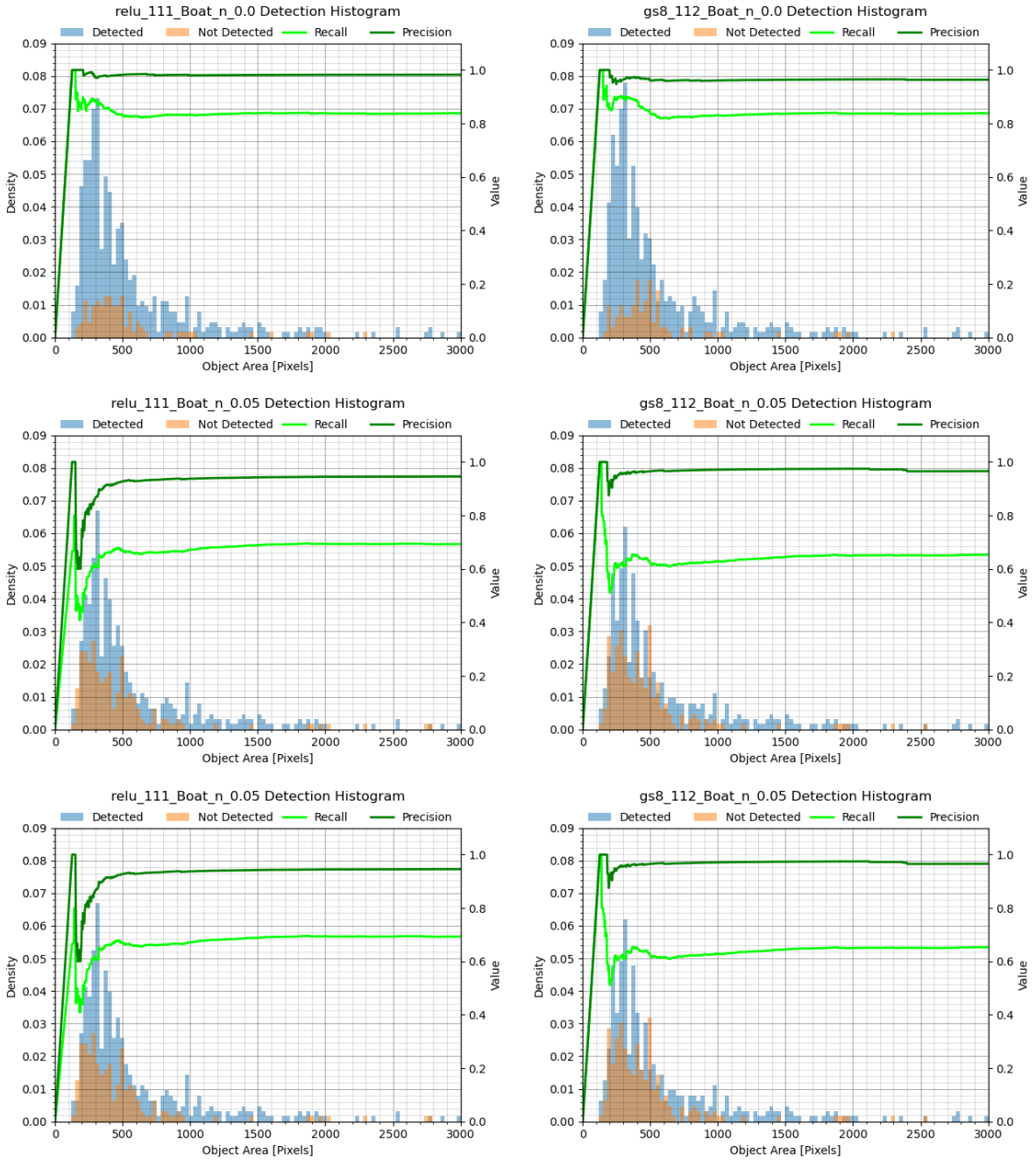


Figure 7. Detection histograms, precision and recall curves as a function of area of boat bounding boxes. The blue histogram is the detected objects where the orange is the objects not detected, dark green curve is the precision and light green is the recall. Comparison of standard ReLU network (figures on the left) versus the "GroupSort 8, SN" network (figures on the right) with no Gaussian noise (top), with $\mathcal{N}(0, 0.05)$ noise (middle) and $\mathcal{N}(0, 0.075)$ (bottom). The left vertical axis show the density of objects of this size in the dataset; the right vertical axis show the percentage of recall/precision and horizontal axis show the bounding box area of each object.

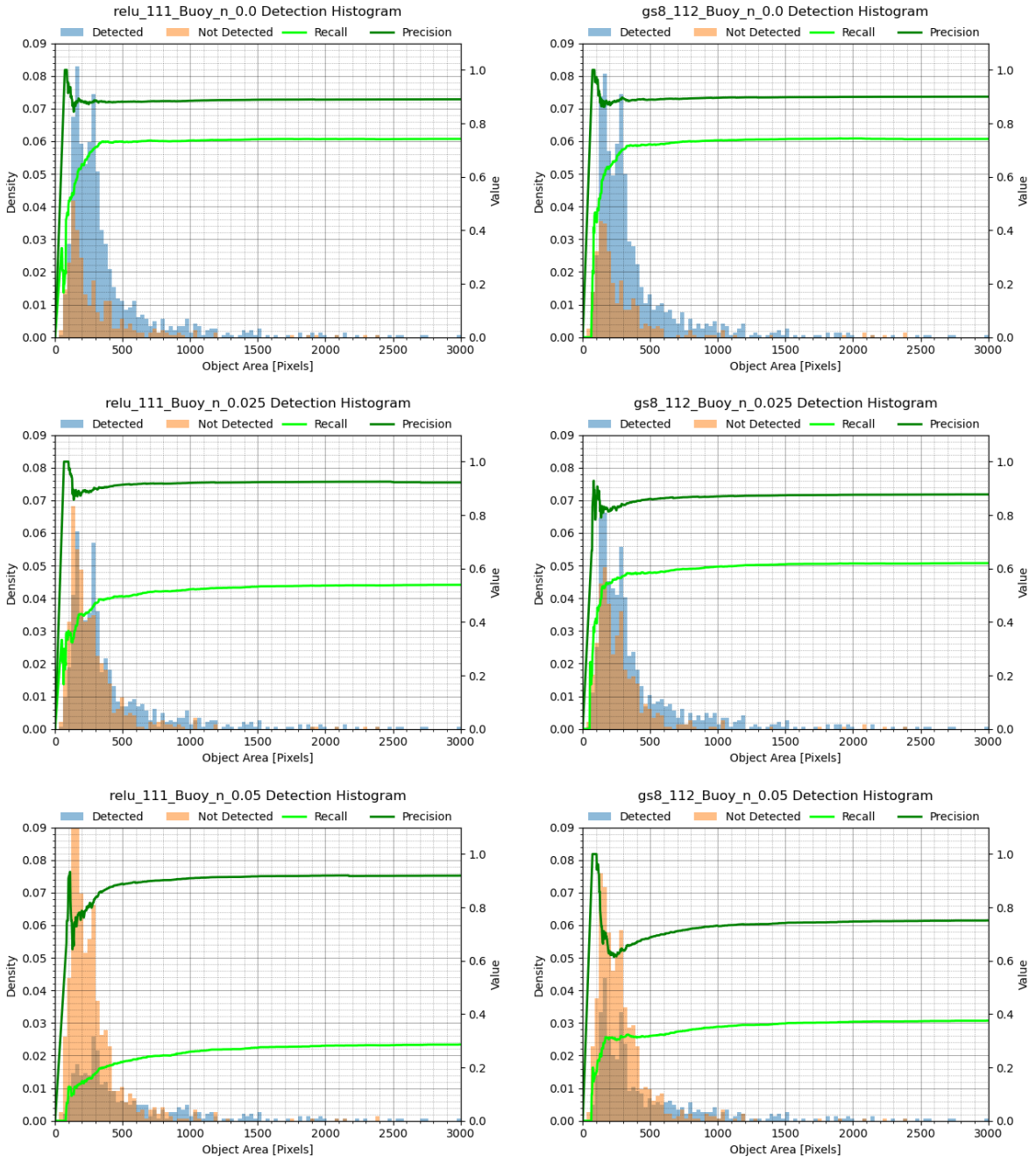


Figure 8. Detection histograms, precision and recall curves as a function of area of buoy bounding boxes. The blue histogram is the detected objects where the orange is the objects not detected, dark green curve is the precision and light green is the recall. Comparison of standard ReLU network (figures on the left) versus the "GroupSort 8, SN" network (figures on the right) with no Gaussian noise (top), with $\mathcal{N}(0, 0.025)$ noise (middle) and $\mathcal{N}(0, 0.05)$ (bottom). The left vertical axis show the density of objects of this size in the dataset; the right vertical axis show the percentage of recall/precision and horizontal axis show the bounding box area of each object.

Figure 9. Images corrupted with different levels of Gaussian blur. From left to right: no noise, $\sigma = 1.5$, $\sigma = 2$.

Figure 10. Images corrupted with different levels of Salt and Pepper noise. From left to right: no noise, 0.5%, 1%.

Table 3. mAP Results for various levels of Salt and Pepper Filter. All results are ±0.02

RetinaNet Model	No Noise	0.1%	0.5%	1%
ReLU	0.76	**0.70**	**0.58**	**0.50**
MaxMin SN	0.75	0.67	0.54	0.48
GS 4, SN	0.76	0.68	0.55	0.50
GS 8, SN	**0.77**	0.69	0.56	0.49

7. Discussion

In this work, Lipschitz constrained network architectures were explored in order to improve general robustness of an object detection network. The architecture used was identical to the one explored in [13], the aforementioned modifications were applied to explore if out of the box modifications could help improve robustness. To get a wider comparison of the modifications early stopping was used and all models were approximately trained to 110 epochs. Training the network to convergence took up to 10 days on a Tesla V100 with 16 GB of memory. In this work, the optimization was not explored to its full extent, as the focus was not the most optimal network but rather how these modifications improve the networks' relative to one another. Given this, it is to be expected that the networks will perform better when trained to convergence.

Most of the tested architecture modifications do not have any negative effect on the zero-noise, uncorrupted data. This fact is very positive, as inferior performance on this base test

scenario, would pose questions about the usefulness of the propose modifications.

Furthermore, as shown in table 1 for the case of Gaussian noise corruption, a noticeable improvement appears once noisy data is used for testing. The standard ReLU network shows a fall in mAP as the level of noise increases, whereas each of the Lipschitz constrained networks do better. Similarly, with Gaussian blur, as shown in table 2, the Lipschitz constrained networks performed better overall.

However, applying Salt and Pepper noise to the input seems to cause a decrease in performance for the GroupSort networks see table 3. A reason for this drop in performance can be the proposed Lipschitz constrained networks. They were conceived to be robust against small specific augmentations of the input which is often the case in adversarial attacks. When the induced noise level is radical, e.g. in Salt and Pepper noise pixel values abruptly change to the highest or lowest possible level of the intensity value range–this type of networks cannot handle the change, resulting in inferior results. The same reasoning can explain the worse performance for the highest tested noise level of Gaussian blur.

For boats, in figure 7, the area of the precision curve of the GroupSort networks is larger as noise is introduced. In the recall curve of the same figure, there is a sharp downward trend with smaller vessels which reduces the total area under the curve, which is not as extreme with the GroupSort networks. With respect to Buoys, see figure 8, as noise is introduced the precision curve is slightly higher on larger vessels, but on smaller vessels it shows that the GroupSort network does better. The area under the recall curve in this figure is also noticeably larger for the GroupSort networks, which translates to a higher percentage of detected buoys.

8. Conclusion & Future Work

In this work, Lipschitz constrained neural networks were explored with the hope of increasing the robustness of object detection at sea. Several Lipschitz constrained modifications were explored with the intent of making a pre-selected network architecture, RetinaNet as used in [13], more robust to noisy input data. The selected modifications, e.g. the GroupSort activation function and Spectral Normalization, ensure the network is Lipschitz continuous. These modified networks have an advantage in the case of Gaussian noise and Gaussian blur and provide only slightly worse results for salt and pepper noise. In all cases though, if the noise extends beyond a certain point the Lipschitz constrained networks break down, whereas the standard ReLU network will still function to some degree.

The obtained results are encouraging, even though not entirely conclusive. For future work, an interesting direction is introducing the explored noise as augmentations of the input for the training of the network. The Lipschitz constrained networks break down when the input is too different from what they have been trained on. If adding augmentations to the input of the network allows the Lipschitz constrained networks to handle bigger variation while still maintaining the higher recall/precision, with respect to small changes it would be very positive. Furthermore some of the Lipschitz modifications were not used due to them being computationally inefficient, such as the Björck convolution. Given additional computational resources, these can be explored as it is only demanding in training time and not in inference time.

Acknowledgments

This research is sponsored by the Danish Innovation Fund, The Danish Maritime Fund, Orients Fund and the Lauritzen Foundation through the ShippingLab project.

References

[1] C. Anil, J. Lucas, and R. Grosse. Sorting out Lipschitz function approximation. In *36th International Conference on Machine Learning, ICML 2019*, volume 2019-June, pages 432–452, 2019.

[2] A. Björck and C. Bowie. Iterative algorithm for computing the best evidence of an orthogonal matrix. *SIAM Journal on Numerical Analysis*, 8(2):358–364, 1971.

[3] A. Chernodub and D. Nowicki. Orthogonal permutation linear unit activation function (OPLU). *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9887 LNCS:533–534, 2016.

[4] M. Cisse, P. Bojanowski, E. Grave, Y. Dauphin, and N. Usunier. Parseval networks: Improving robustness to adversarial examples. *34th International Conference on Machine Learning, ICML 2017*, 2:1423–1432, 2017.

[5] R. Dulski, S. Milewski, M. Kastek, P. Trzaskawka, M. Szustakowski, W. Ciurapinski, and M. Zyczkowski. Detection of small surface vessels in near, medium, and far infrared spectral bands. In *Electro-Optical and Infrared Systems: Technology and Applications VIII*, 2011.

[6] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016.

[7] D. P. Kingma and J. L. Ba. ADAM: A method for stochastic optimization. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, pages 1–15, 2015.

[8] T. Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017.

[9] T. Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar. Focal Loss for Dense Object Detection. *Proceedings of the IEEE International Conference on Computer Vision*, pages 2999–3007, 2017.

[10] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida. Spectral normalization for generative adversarial networks. In *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings*. International Conference on Learning Representations, ICLR, 2018.

[11] L. Nalpantidis and A. Gasteratos. Stereo vision for robotic applications in the presence of non-ideal lighting conditions. *Image and Vision Computing*, 28(6):940–951, 2010.

[12] F. E. T. Schöller, M. Blanke, M. K. Plenge-Feidenhans'l, and L. Nalpantidis. Vision-based object tracking in marine environments using features from neural network detections. In *IFAC Worls Congress*, Berlin, Germany, 2020.

[13] J. D. Stets, F. E. T. Schöller, M. K. Plenge-Feidenhans'l, R. H. Andersen, S. Hansen, and M. Blanke. Comparing Spectral Bands for Object Detection at Sea using Convolutional Neural Networks. *Journal of Physics: Conference Series*, 1357:12036, oct 2019.

[14] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. In *2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings*, 2014.