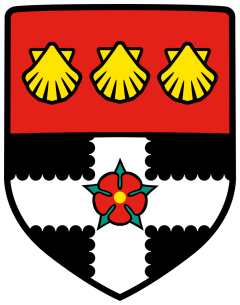
Class-agnostic maritime object detection



Thomas Antony Cane

Department of Computer Science
University of Reading

This dissertation is submitted for the degree of
Doctor of Philosophy

November 2019

To Barry

Declaration

I hereby declare that, except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this or any other university. I confirm that this dissertation is my own work and the use of material from other sources has been properly and fully acknowledged.

Thomas Antony Cane
November 2019

Acknowledgements

First of all, I'd like to thank my supervisor, James, for his advice and guidance in putting this research together and keeping it (and me) on track over the last 5 years. I am also hugely grateful to everyone in the Computational Vision Group for their discussions and ideas. In particular, I'd like to thank Luis, Jonathan and Chris for their help with wrangling data, attaching cameras to boats in France, and deciding how much to bet in casinos in Greece.

My thanks go to all the partners and collaborators in the IPATCH project. I'd like to especially thank Jörgen, Amanda and Per-Magnus at TST for their collaboration on the system evaluation and use of the MTT software.

My employer, BMT, kindly sponsored my studies and allowed me time off work to complete the write up. I'm especially grateful to Rory, without whose support this PhD would not have happened. Special thanks go to the Research team for sharing the horror stories from their own PhD experiences...!

To all my family and friends – thank you for your understanding when I had to miss social occasions or had to drop out of things last minute. I am very much looking forward to catching up with you all and finding out what you have been doing over the last five years.

Finally, to my ~~girlfriend~~, ~~fiancée~~, wife, Nicola – thank you for the support, cups of coffee, shoulder rubs, horizon groundtruthing, proofreading and, above all, patience. Sorry it's taken over our life somewhat, but thank you for helping me get to the end, just in time for an even bigger adventure...

Abstract

This thesis proposes new computational methods for detecting maritime objects in video data and analyses their performance in the context of a counter-piracy surveillance system. A key challenge in the maritime anti-piracy context is the wide range of possible environments and objects which may be encountered. The focus of this work is therefore on methods which do not make strong assumptions about the visual appearance of the scene or targets. This has the additional benefit that they can be used in other applications and domains.

Two novel approaches are investigated. The first is a saliency-based method which uses a novel thresholding step and a scene depth map derived from the horizon to emphasise local saliency and incorporate scene context, respectively. The second uses a deep semantic segmentation network to separate 'sea', 'sky' and 'other' regions. Contextual scene knowledge is then used to extract objects by applying rule-based reasoning. Evaluation on publicly available maritime surveillance datasets shows that the proposed methods address limitations of current approaches, particularly with regard to the detection of small, distant targets.

The analysis also explores the key aspects of performance required to deploy an algorithm 'in the field' as part of a larger system for detection, tracking, situation awareness and threat detection. As well as the trade-off between real-time operation and performance, results on data collected from a real-world surveillance system show that the relationship between detection scores in images and tracking performance in the real-world is not trivial.

Finally, contributions are made in the benchmarking and evaluation of image-based maritime object detection methods, including a novel dataset for counter-piracy surveillance, improvements to metrics for performance evaluation in the maritime domain, and comparison of the proposed approaches with state of the art object detection methods from other domains.

Table of contents

List of figures **xv**

List of tables **xxi**

1 Introduction **1**

1.1 Motivation . . . 1
1.2 Challenges . . . 3
1.3 Objectives . . . 5
1.4 Contributions . . . 6
1.5 Related publications . . . 6
1.6 Outline of this thesis . . . 7

2 Related Work **9**

2.1 Introduction . . . 9
2.2 Object detection in maritime surveillance . . . 10
2.2.1 Background subtraction . . . 10
2.2.2 Foreground segmentation . . . 11
2.2.3 Object models . . . 11
2.2.4 Deep learning . . . 12
2.3 Saliency, visual attention and salient object detection . . . 12
2.3.1 Saliency and visual attention-based approaches in the maritime domain . . . 13
2.3.2 Salient object detection approaches . . . 17
2.4 Deep learning for semantic segmentation . . . 17
2.5 Maritime object detection in real-world systems . . . 20
2.6 The IPATCH project . . . 22

2.7 Summary . . . 22

3 Evaluation and Benchmarking 25

3.1 Introduction . . . 25
3.2 Object detection and tracking . . . 25
3.2.1 Datasets and challenges . . . 25
3.2.2 Performance evaluation . . . 26
3.3 Maritime surveillance datasets . . . 27
3.3.1 Publicly available datasets . . . 28
3.3.2 Non-public datasets . . . 35
3.3.3 Synthetic datasets . . . 35
3.3.4 Summary and technical challenges . . . 36
3.4 Metrics for maritime surveillance . . . 39
3.5 Evaluation methodology for maritime object detection in this thesis . . . 48
3.5.1 Selection of sequences and groundtruth for object detection evaluation 48
3.5.2 Evaluation metrics used in this thesis . . . 48
3.5.3 Practical upper bound on performance . . . 54
3.6 Baseline methods and performance . . . 57
3.6.1 Methods . . . 57
3.6.2 Selection and configuration of deep object detection network variants 59
3.6.3 Performance evaluation . . . 69
3.6.4 MODP vs. MODP-GT$_{BEP3}$ . . . 75
3.7 Improved Mean Absolute Error for evaluating saliency maps . . . 76
3.7.1 Mean Absolute Error (MAE) . . . 76
3.7.2 Shortcomings . . . 76
3.7.3 Proposed improvements . . . 78
3.8 Summary . . . 80

4 Visual Attention and Saliency for Object Detection 81

4.1 Introduction . . . 81
4.2 Evaluation on maritime surveillance data . . . 82
4.2.1 Experimental set-up . . . 82
4.2.2 Results and analysis . . . 89
4.3 Creating an object detector for maritime surveillance . . . 102

4.3.1 Boolean Map Saliency . . . 102
4.3.2 Candidate region extraction . . . 105
4.3.3 Mitigating saliency of wake through horizontal and vertical thresholding . . . 108
4.3.4 Reducing false positives through temporal filtering . . . 108
4.4 Incorporating scene context through horizon detection . . . 112
4.4.1 Horizon detection . . . 112
4.4.2 Depth-weighted activation maps in BMS . . . 120
4.4.3 Creating a depth map from the horizon . . . 121
4.5 Evaluation and comparison against baselines . . . 127
4.5.1 Experimental set-up . . . 127
4.5.2 Results and analysis . . . 129
4.6 Summary . . . 139

5 Semantic Segmentation for Object Detection 141
5.1 Introduction . . . 141
5.2 Semantic segmentation networks . . . 143
5.2.1 Selected networks . . . 143
5.2.2 Baseline performance on CamVid dataset . . . 143
5.3 Training on data from the ADE20k dataset . . . 154
5.3.1 Training data . . . 154
5.3.2 Implementation and training . . . 156
5.4 Experiments . . . 161
5.4.1 Number of classes . . . 161
5.4.2 Data augmentation . . . 165
5.4.3 Multi-task learning . . . 168
5.4.4 Incorporating global spatial information . . . 172
5.4.5 Results and analysis . . . 173
5.5 Creating an object detector for maritime surveillance . . . 180
5.6 Evaluation and comparison against baselines . . . 184
5.6.1 Experimental set-up . . . 184
5.6.2 Results and analysis . . . 184
5.7 Summary . . . 193

6 Real-World Performance Evaluation **195**
6.1 Introduction . . . 195
6.2 On-board surveillance system . . . 196
6.2.1 Overview . . . 196
6.2.2 Early Detection Module . . . 197
6.2.3 Multi-target tracking module . . . 198
6.3 Real-world set-up and data collection . . . 200
6.3.1 System deployment . . . 200
6.3.2 Calibration and synchronisation . . . 203
6.4 Experiments . . . 205
6.4.1 Implementation . . . 205
6.4.2 Sequences . . . 207
6.4.3 Evaluation procedure . . . 208
6.4.4 Sources of error . . . 215
6.4.5 Summary of MTT inputs . . . 216
6.5 Results and analysis . . . 218
6.5.1 System error . . . 218
6.5.2 Performance using object detection methods . . . 218
6.5.3 Comparison with radar . . . 220
6.5.4 Comparison with thermal cameras . . . 226
6.6 Summary . . . 228

7 Conclusions and Future Work **229**
7.1 Findings and limitations . . . 229
7.1.1 Saliency-based object detection . . . 229
7.1.2 Semantic segmentation-based object detection . . . 230
7.1.3 Visual object detection in the real world . . . 231
7.2 Outcomes against objectives . . . 231
7.3 Future work . . . 233

References **235**

Appendix A Data Augmentations **253**

List of figures

1.1 Cameras of a prototype visual surveillance system for protecting ships against piracy . . . 2
1.2 A typical surveillance pipeline . . . 3
1.3 Structure of this thesis . . . 7

2.1 Itti & Koch computational model . . . 14
2.2 Boolean map saliency block diagram . . . 15

3.1 Example MarDCT images . . . 29
3.2 Example MODD images . . . 29
3.3 Example MARVEL images . . . 30
3.4 Example MASATI images . . . 31
3.5 Example PETS 2005 images . . . 31
3.6 Example IPATCH images from the 2015, 2016 and 2017 campaigns . . . 32
3.7 Example Singapore Maritime Dataset images . . . 32
3.8 Example SEAGULL images . . . 33
3.9 Example SeaShips images and detection results . . . 34
3.10 Example VAIS images . . . 34
3.11 Example SMARTEX images . . . 35
3.12 Technical challenges for visual detection in the maritime environment . . . 37
3.13 Characteristics of common maritime targets and issues with bounding box-based evaluation . . . 40
3.14 Importance of detecting hull accurately, compared to upper structures. . . . 41
3.15 Notation for the bottom-edge precision (BEP) metrics . . . 42
3.16 Analysis of Prasad BEP metrics . . . 43
3.17 Specific cases and how BEP1 and BEP3 compare to the intuitive ranking . . . 44

3.18 Example showing the X and Y components of BEP3 plotted on a scatter graph 46
3.19 Example showing the X and Y components of BEP3 plotted against frame number 47
3.20 Example showing effect of BEP3 version compared to MODP 53
3.21 Examples of challenging objects to annotate 55
3.22 Tolerance of DR curve to 1px perturbations for 4 IPATCH sequences with increasing numbers of small object frames 56
3.23 Tolerance of MODP-GT to 1px perturbations for 4 IPATCH sequences with increasing numbers of small object frames 56
3.24 Tolerance of MODP-GT to perturbations with increasing object size within a sequence 56
3.25 Examples showing the contrast in the type of target objects from the IPATCH and SMD datasets 61
3.26 Mean METE for different confidence and NMS thresholds across all sequences 63
3.27 Mean METE for different confidence and NMS thresholds across 3 SMD sequences 64
3.28 Mean METE for different confidence and NMS thresholds across 4 IPATCH sequences 65
3.29 Average processing speed per frame for the deep network variants 68
3.30 Qualitative results for the baseline methods 71
3.31 Detection rate curves for the baseline methods 72
3.32 MODP-BEP3 results for the baseline methods on IPATCH sequences 73
3.33 MODP-BEP3 results for the baseline methods on SMD sequences 73
3.34 MODP-BEP3 results for the baseline methods on SEAGULL sequences . . . 74
3.35 Average processing speed per frame for the baseline methods 74
3.36 Effect of using MODP-BEP3 instead of MODP for evaluation 75
3.37 MAE behaviour for different strategies as a function of object size 77
3.38 Behaviour of the proposed Balanced MAE metric under different strategies 79

4.1 Block diagram showing the saliency-based object detection concept 82
4.2 Example images from the sub-sequences for evaluating saliency performance 83
4.3 Examples of groundtruth saliency maps created for the sub-sequences in Table 4.1 85
4.4 Example saliency maps from biologically-inspired methods 91

4.5 Example saliency maps from frequency analysis methods 92
4.6 Example saliency maps from salient object detection methods 93
4.7 $\widehat{\text{BMAE}}$ vs. frame number for biologically-inspired approaches 95
4.8 $\widehat{\text{BMAE}}$ vs. frame number for spectrum analysis approaches 96
4.9 $\widehat{\text{BMAE}}$ vs. frame number for salient object detection approaches 97
4.10 $\widehat{\text{BMAE}}$ vs. frame number for the best approaches from each category 98
4.11 PR and ROC curves for biologically-inspired approaches 99
4.12 PR and ROC curves for spectrum analysis approaches 99
4.13 PR and ROC curves for salient object detection approaches 100
4.14 PR and ROC curves for the best approaches from each category 100
4.15 Block diagram presenting the different stages of the proposed algorithm. . . 102
4.16 Stages of the BMS pipeline 103
4.17 Example maps from each stage of the BMS pipeline 105
4.18 Example images, BMS saliency maps and saliency value distributions for a salient object detection benchmark image and a maritime surveillance image 106
4.19 Distributions of object size for the MSRA-B dataset 107
4.20 Visualisation of horizontal-vertical thresholding 109
4.21 Examples of horizon detection failure using the region separation method . 114
4.22 Behaviour of region separation score based on horizons detected by the Park method 115
4.23 Selected horizon line detection method from Park et al. [158] 116
4.24 Example horizon detection results on IPATCH sequences 117
4.25 Effect of using scene depth to weight the activation maps in BMS 122
4.26 Horizon geometry 123
4.27 Horizon-based depth map functions 125
4.28 Example depth map cross section 126
4.29 Integration of horizon detection in the saliency method 128
4.30 Effect of different thresholding values on the SMD-1615 and IPATCH 2015-Sc2a_Tk1-CAM11 sequences 130
4.31 Effect of different thresholding values on MODP-BEP3 for the SMD-1615 and IPATCH 2015-Sc2a_Tk1-CAM11 sequences 131
4.32 Effect of horizontal-vertical thresholding on SMD and IPATCH sequences . 132
4.33 Effect of horizontal-vertical thresholding on wake and reflections 133

4.34 MODP-BEP3 vs. frame number for IPATCH sequences where targets approach from a large distance . . . 134
4.35 Effect of temporal filtering on false positives and MODP-BEP3 performance 136
4.36 Speed vs. performance trade-off . . . 137
4.37 Comparison of the proposed saliency-based object detection method with baseline methods from the literature. . . . 138

5.1 The semantic segmentation-based object detection concept . . . 142
5.2 Example CamVid [41] images and groundtruth labels . . . 145
5.3 CamVid baseline training and validation curves . . . 148
5.4 Semantic segmentation network properties based on CamVid baseline experiments . . . 150
5.5 CamVid baseline validation per-class accuracies and IoUs . . . 151
5.6 Inference speeds for the networks on different image sizes . . . 151
5.7 Qualitative results for the networks on CamVid in the baseline experiment. 152
5.8 Example training images and class mappings from the 434-image maritime subset of ADE20k [248]. . . . 155
5.9 k-fold cross-validation training for 3-plus-1 mapping . . . 158
5.10 Examples of MarSemSeg images and groundtruth labels . . . 159
5.11 Results of the 4 networks on MarSemSeg test set, trained on full ADE20k maritime subset . . . 162
5.12 Inference time on MarSemSeg . . . 162
5.13 Confusion matrices for the 4 networks on MarSemSeg test set, trained on full ADE20k maritime subset . . . 163
5.14 Example segmentation output for the networks on MarSemSeg . . . 164
5.15 Ablation study results . . . 167
5.16 Implementation of multi-task learning by adding an extra output channel to the network and applying separate losses. . . . 168
5.17 Example horizon map examples. . . . 170
5.18 Example boundary map. . . . 170
5.19 CoordConv concept . . . 172
5.20 Training loss and mIoU for the baseline variants . . . 174
5.21 Training loss and mIoU for the multi-task variants . . . 174

5.22 Training curves for horizon prediction (a-b) and boundary prediction (c-d) multi-task training. 175
5.23 Global accuracy and mIoU results for the semantic segmentation network variants on MarSemSeg 176
5.24 Per-class accuracy and IoU results for the semantic segmentation network variants on MarSemSeg 177
5.25 Qualitative results showing effect of data augmentation 178
5.26 Qualitative results showing effect of CoordConv 178
5.27 Qualitative results showing effect of the horizon prediction and boundary prediction tasks 179
5.28 Qualitative results showing effect of the boundary prediction task and CoordConv 179
5.29 The proposed semantic segmentation-based object detection method . . . 180
5.30 Scene reasoning decision process 182
5.31 Output of the scene reasoning decision process compared to argmax class predictions 183
5.32 Relationship between segmentation performance and object detection performance. 185
5.33 MODP-BEP3 results for the semantic segmentation-based object detection method on the IPATCH sequences. 187
5.34 MODP-BEP3 results for the semantic segmentation-based object detection method on the SMD sequences. 188
5.35 MODP-BEP3 results for the semantic segmentation-based object detection method on the SEAGULL sequences. 189
5.36 Example of consistent good performance on challenging targets (baseline-coordconv on IPATCH 2016-Sc1_Tk5-CAM11) 189
5.37 Examples of good performance and limitations on SMD sequences. 190
5.38 Examples comparing horizon and boundary prediction (a, d) and CoordConv (b, c, e, f) on IPATCH sequences. 190
5.39 Comparison of the proposed semantic segmentation-based object detection method with the saliency-based method from Chapter 4 and baseline methods from the literature. 192

6.1 High-level architecture of the IPATCH on-board surveillance system 197

6.2 High-level architecture of the Early Detection Module 198
6.3 Screenshot of the multi-target tracker (MTT) [4] 199
6.4 Ship-centred coordinates (SCC) 199
6.5 Region and vessel used in the 2015 and 2016 campaigns 201
6.6 Camera installations on the VN Partisan 202
6.7 Target skiffs from the 2015 and 2016 trials 202
6.8 Region and vessel used in the 2017 campaign 203
6.9 Camera installation on the Kamari 204
6.10 Target skiffs from the 2017 trials 204
6.11 Relationship of work in this thesis to the IPATCH system 206
6.12 Effect of smoothing on roll and pitch estimation from horizon detection (Scenario 2016-Sc1_Tk5-CAM11) 207
6.13 Plots of the three scenarios used to evaluate the object detection methods in a real-world context 209
6.14 Skiffs in 2017-Sc3a appearing on the radar screen on-board the Kamari . . . 210
6.15 Illustration of the timestep binning and alignment concept 212
6.16 Absolute (left) and relative (right) errors for GPS and bounding box inputs . 218
6.17 MTT outputs for scenario 2015-Sc3_Tk2 220
6.18 MTT outputs for scenario 2016-Sc1_Tk5 222
6.19 MTT outputs for scenario 2017-Sc3a 223
6.20 Tracks using the radar detections as input to the MTT 224
6.21 Tracks from the thermal cameras for 2017-Sc3a 226

List of tables

2.1 Visual attention and saliency based methods in the maritime domain 15

3.1 Challenges present in maritime datasets 36
3.2 Summary of datasets used in this thesis 38
3.3 Sequences selected for object detection evaluation 49
3.4 Baseline deep object detection variants and reported performance on the COCO dataset 59
3.5 Baseline mean and median MODP-BEP3 results for the Mask R-CNN and YOLO variants 67
3.6 Baseline method FAF 70

4.1 Sub-sequences for saliency map evaluation 84
4.2 Summary of saliency method implementations 88
4.3 BMAE, AUC PR and AUC ROC results for all methods 89
4.4 BMAE, AUC PR and AUC ROC results for the top performing methods from each category 101
4.5 Results for horizon detection methods on different datasets 118
4.6 Processing speed (mean ms/frame) for horizon methods 119
4.7 Key to saliency method variants 128

5.1 Semantic segmentation networks shortlist 144
5.2 Semantic segmentation network training setup 146
5.3 Reported vs. achieved results for CamVid baseline test 148
5.4 Dataset properties for the 3 class mappings of the maritime subset of ADE20k 156
5.5 Dataset properties for the 3 class mappings of MarSemSeg 160
5.6 Key to semantic segmentation method variants (all use the EDANet network) 173

5.7 False positive results (FAF) for the proposed and baseline methods on representative IPATCH sequences. 191

6.1 Summary of detection sources processed by the MTT 217
6.2 Sequence-level results for the visual object detection methods 221
6.3 Time to detection, measured as the first timestep with a True Positive track detection. 222
6.4 Sequence-level results comparing radar against the visual object detection methods 225
6.5 Comparison of visual object detection methods with object detections from thermal cameras 227

Chapter 1

Introduction

1.1 Motivation

The use of automated video analytics is becoming commonplace in urban and indoor environments where they are used for detection and tracking of pedestrians and vehicles. In the maritime domain, there is great potential for exploiting similar systems for monitoring the sea for the purposes of safety and security. However, the use of automated vision-based surveillance methods in maritime environments remains immature compared to other domains.

Maritime surveillance is important for situational awareness in a range of applications to ensure the safety and security of vessels, ports and other maritime infrastructure. Radar is the *de facto* standard for object detection and tracking in the maritime domain and is widely deployed on land and on vessels. AIS (Automatic Identification System) is also widespread, as it is mandatory under international legislation [98] for vessels over 300 Gross Tonnage and all passenger ships. Under the system, ships report their identity, position, speed and other characteristics over VHF in a standard message format. Radar and AIS are often used together but the information is still limited compared to the rich data offered by visual sensors.

In most cases, a human operator still represents the state of the art for gathering visual information from a maritime scene, whether it's from land or from on board a vessel. Cameras are commonly installed in such locations, so automated video analytics is an attractive option for extending maritime surveillance systems. On the other hand, the maritime environment presents additional technical challenges for computer vision



Fig. 1.1 Cameras of a prototype visual surveillance system for protecting ships against piracy

methods and real-world systems, so this is perhaps why their use is less widespread than in indoor and urban environments.

After a resurgence in 2011, maritime piracy continues to place a huge economic and human cost on commercial shipping around the world [153]. The most effective protection for ships is a proper lookout to maximise early warning of a potential attack, allowing time for the crew to prepare accordingly [100]. Radar and crew members with binoculars represent the state of the art available to commercial fleets[1]. However, the navigation radar available on ships does not perform well with small, fast-moving objects [218] such as the 'skiffs' used by pirates, and crew members become fatigued after maintaining a lookout for a long period. Automated visual surveillance offers a new sensing modality for ships which could operate continuously without human intervention and increase the early detection of piracy threats.

Many of these difficulties can be addressed by using thermal cameras. Unfortunately, the cost of thermal sensor hardware is prohibitive for many applications and the technology is restricted for civilian use. Visible light cameras offer a more affordable alternative which could provide surveillance coverage of a larger region and complement other available sensors, such as radar. This motivates further research into improving their performance for operation in maritime environments. Finally, if methods for detecting objects in maritime environments using visual cameras can be perfected, there are many

[1] AIS is of no use in this case, as pirates do not tend to broadcast their position



Fig. 1.2 A typical surveillance pipeline

applications where the capability would have value. Navigational safety is an on-going concern for the maritime industry. Vessel traffic monitoring services in ports and harbours, and crews on-board vessels would benefit greatly from enhanced situational awareness. Autonomous ships are also on the horizon and visual detection and tracking systems are likely to play a big role in obstacle avoidance and navigation [128].

1.2 Challenges

In the context of vision-based maritime surveillance, an ideal object detector:

- is able to process high quality video fast enough for real-time operations (for example >15Hz)
- locates objects in the image accurately enough for higher-level tasks in the surveillance pipeline to perform their function well, such as situational awareness and threat detection (see Fig. 1.2)
- has a low false positive rate
- is performant over a wide range of viewpoints, object types and environmental conditions

The technical challenges commonly associated with visual detection – varying illumination, object occlusion, and so on – are present in maritime surveillance applications but the nature of environments on or near the sea can magnify them. A method that is robust to small illumination changes in an indoor environment, for example, may not perform well when faced with the diverse levels of scene illumination that can exist at sea. To compound the problem further, the maritime environment presents additional challenges which are unique to the sea.

Maritime scenes range from constrained, well-defined areas of water to open, unconstrained expanses of ocean. Constrained environments include ports, harbours and bays. Unconstrained environments exist where there is nothing but water and sky in the field of

view. This occurs when the platform is in the open sea (i.e. all land is beyond the visible horizon), but also when looking out from the shore away from land (e.g. monitoring coastal/littoral regions).

The choice of surveillance platform and location has a big influence on the appearance of maritime scenes and objects within them. Viewpoints could range from cameras mounted on a floating buoy a few feet above the surface of the water, to cameras carried by aircraft or drones a few hundred metres in the air. In this research, the problem is constrained to cameras which are mounted on the shore (land-based), on airborne vehicles (aerial), or on surface vessels (vessel-based). Remote sensing (satellite observation) is not included because of the vastly different distance scales involved and because they operate on images, rather than video. Detection is possible (and satellites are indeed used for this purpose) but real-time surveillance is not practical, as the satellite must complete its orbit before the same area of sea can be observed again, leading to a large time lag between observations.

Maritime scenes have highly dynamic backgrounds, making it difficult to create reliable models for background subtraction. Waves create highly reflective surfaces which move through the scene and the tops of waves often form white foamy cusps which appear as small, high-contrast regions in the image. Waves and foam are also created in the wake of boats. There is therefore a lot of clutter in the sea region of the scene, which poses a challenge for false positives performance. The other major element of maritime images is the sky. This may be completely empty or filled with clouds that constantly move and change shape. These in turn create a constantly varying illumination of the scene so algorithms must be robust to this.

The weather at sea is far more variable than on land, so environmental conditions can change considerably on small timescales and tend to be more extreme. This introduces significant variation in the appearance of both the background and the objects of interest during system operation. Rain, haze and fog – unfortunately common at sea – absorbs and disperses light reflected from objects, which reduces their contrast against the background. Different wavelengths are also absorbed at different rates, which can lead to colour variation.

The characteristics of targets in maritime scenes create further challenges. Their size can range from that of a swimmer or jetski (a few metres) up to the length of a large tanker or container ship (a few hundreds of metres). Compared to indoor and urban

environments, where objects of interest are people and vehicles, the range in size is much larger. Similarly, their speed can range from completely stationary (i.e. moored in harbour or at anchor) up to 20 kn for a shipping vessel or 60 kn or more for a fast private yacht. Because of the long distances that are observable at sea (due to an absence of buildings or landscape features limiting the view), objects can appear in the field of view as close as 10 m and as far as the horizon (around 11 km for an observer height of 10 m). These factors mean that a small, close object could appear very similar to a large, distant object from the camera's point of view. Detection, tracking and classification systems must be able to perform consistently across a large range of image scales.

The unconstrained nature of the sea also means that it is difficult to limit problems to a subset of object types, as potentially any type of object could come into the field of view (although reasonable assumptions are usually made about this). Not only that, but because objects may be approaching from anywhere in the scene (including behind the camera), the orientation of the target with respect to the camera could be anything from 0° to 360°. This is particularly problematic with long vessels as their profile changes considerably depending on whether they are viewed from the side or the end.

1.3 Objectives

Maritime surveillance is a broad problem which presents varied challenges. The focus of this thesis will be on addressing the anti-piracy use case. The main objectives of this research are to develop object detection methods which

- can detect small, fast-moving skiffs approaching the vessel as early as possible to maximise warning for the crew
- provide high quality detections to the higher-level stages of an on-board piracy surveillance system to support tracking, situational awareness and threat detection
- do not make strong assumptions about the appearance of the target or scene so that they can be used in a wide range of contexts and applications
- are robust to camera motion, wake, reflections and environmental conditions
- can operate in real-time

To support this, a further objective is to evaluate and compare the performance of the proposed methods and others from the literature in the context of a real-world maritime surveillance system using realistic data.

1.4 Contributions

The main contributions of this thesis are:

- Two novel class-agnostic object detection methods for maritime surveillance, one using visual saliency and scene context, and the other using semantic scene segmentation and rule-based reasoning
- Improved maritime-oriented performance evaluation metrics, building on work by Prasad et al. [167], and analysis of their behaviour under different conditions
- Improvements to the Mean Absolute Error measure, commonly used in evaluating saliency map quality, to reduce the dependence on object size
- Performance evaluation of object detection methods in the context of a real-world maritime surveillance system for protection of ships against piracy

1.5 Related publications

The work in this thesis is linked to a number of co-authored publications. Chapter 4 and 5 incorporate and extend work from:

- T. Cane and J. Ferryman, "Saliency-based detection for maritime object tracking", *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition Workshops*, 2016

- T. Cane and J. Ferryman, "Evaluating deep semantic segmentation networks for object detection in maritime surveillance", *Proceedings of the IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, 2018

At the start of this research, contributions were made to an algorithm already under development. This algorithm is used as one of the baseline methods in the comparisons:

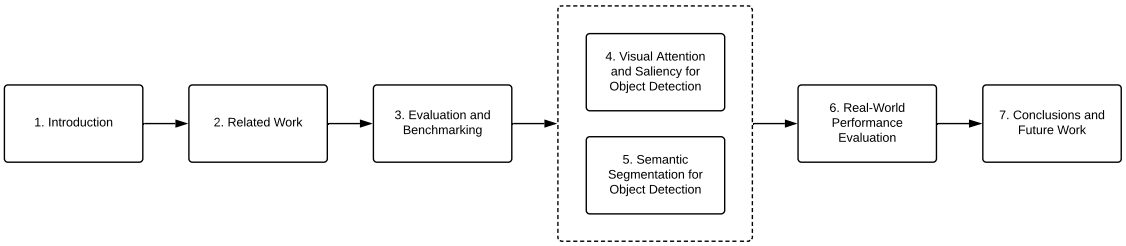


Fig. 1.3 Structure of this thesis

- C. Osborne, T. Cane, T. Nawaz and J. Ferryman, "Temporally-stable feature clusters for maritime object tracking in visible and thermal imagery", *Proceedings of the IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, 2015

Several publications were made as part of the IPATCH project [102] relating to the dataset and on-board system:

- L. Patino, T. Cane, A. Vallée and J. Ferryman, "PETS 2016: Dataset and challenge", *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2016
- L. Patino, T. Nawaz, T. Cane and J. Ferryman, "PETS 2017: Dataset and challenge", *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2017
- M. Andersson, R. Johansson, K-G. Stenborg, R. Forsgren, T. Cane, G. Taberski, L. Patino and J. Ferryman, "The IPATCH System for Maritime Surveillance and Piracy Threat Classification", *Proceedings of the 2016 European Intelligence and Security Informatics Conference (EISIC)*, 2016

1.6 Outline of this thesis

The rest of this thesis is structured as follows (see also Fig. 1.3). Chapter 2 reviews related work to provide background and context to the contributions. Chapter 3 reviews evaluation and benchmarking practices for object detection and identifies issues when applying these metrics in the maritime surveillance context. To address these, new metrics are proposed which are tailored for evaluating maritime object detection. The datasets

used for the experiments are introduced, and the evaluation methodology for the rest of the chapters is laid out.

Chapter 4 and 5 represent the core contributions from this thesis, building on published work [44, 45]. Chapter 4 takes inspiration from the human vision system and applies the idea of visual attention and saliency to the maritime object detection task. Visual attention and saliency methods from the literature are analysed on maritime data and the most promising is adapted into an object detection mechanism. Horizon detection is also introduced as a way of incorporating scene context into the object detection process.

Chapter 5 inverts the object detection approach by segmenting the whole scene into its semantic classes – primarily sea and sky for the maritime case – and using this information to find regions which are neither sea nor sky. Deep neural networks are used for the semantic segmentation step. As training data for scene segmentation is severely limited in the maritime domain, a related dataset is adapted for the purpose and various training mechanisms are investigated to improve the ability of the networks to generalise to a a different data domain.

In Chapter 4 and 5, the proposed object detection methods are evaluated using image-based metrics and compared against baseline methods from the literature. In Chapter 6, the proposed and baseline methods are evaluated in the context of a real-world piracy surveillance system by feeding their object detections into a multi-target tracking module. The quality of the tracks using each input is assessed and compared against tracks from radar and thermal cameras. Finally, Chapter 7 consolidates the conclusions and observations from all the chapters and proposes extensions and further research for the future.

Chapter 2

Related Work

2.1 Introduction

This chapter reviews related work to provide background and context for the main contributions. Object detection is a very broad field, so the first section focusses on approaches which have been applied in the maritime domain. These are summarised under four main categories.

The work in Chapter 4 draws on work in the related fields of saliency, visual attention and salient object detection. These concepts are defined and the key methods are presented to explain why this approach was selected for further research in the maritime context. Similarly, in Chapter 5, the proposed approach builds on previous research in deep neural networks for semantic segmentation. The key advancements and architectures from this area are discussed to justify the choice of these techniques for maritime object detection.

A key objective of this work is to analyse the performance of detection methods. The related work in benchmarking and performance evaluation is saved for Chapter 3, but this chapter provides background on other real-world applications in the literature, and commercial systems which report the use of vision-based detection. Finally, the IPATCH project is introduced, as this forms the backdrop for the research and is the source of the piracy dataset used in the experiments.

2.2 Object detection in maritime surveillance

Generally, the methods deployed in maritime object detection are the same techniques that are used for detecting objects in other contexts, such as pedestrians, faces and vehicles. Historically, this has included milestone works such as classifier cascades [217], histograms of oriented gradients (HOG) [57] and deformable parts models (DPM) [73]. The most recent step change has come from deep learning. A comprehensive review of the evolution of object detection over the last 20 years has recently been published [249]. In the following sub-sections, the approaches that have been explored in the maritime domain are summarised under several main categories.

2.2.1 Background subtraction

Background subtraction is commonly used in situations where the camera is fixed and illumination changes are minimal or predictable. However, in maritime scenes, the background is highly dynamic due to the motion of the sea and in many applications the camera is also mobile (e.g. mounted on a vessel or buoy). With this in mind, it's somewhat surprising that background subtraction is such a popular choice in the maritime domain. Approaches include frame differencing and averaging [181, 219, 221, 243, 244], thresholding [35, 72, 142, 169, 170, 233], probabilistic models [1, 23, 60, 76, 110, 112, 192, 208], and spatiotemporal models [31, 48, 202, 203, 229]. A comprehensive review and performance evaluation of maritime background subtraction methods was conducted recently [164].

A notable method is work by Bloisi et al. on the Independent Multimodal Background Subtraction (IMBS) algorithm [24, 28, 30]. This method states it is designed specifically to address the challenges of dynamic maritime backgrounds. It is therefore selected as one of the baseline methods for comparison in this study.

A key challenge with using background subtraction methods with a dynamic background is balancing the different time scales of motion in the scene. Maritime scenes make this very difficult to achieve in practice. For mobile cameras, the sea causes motion of the whole scene. The nature and magnitude of the motion depends on the sea conditions and the size of platform on which the camera is mounted (for example, a small USV has large, rapid motion, whereas a large tanker exhibits slower and smaller movements).

Stationary and slow moving objects present another challenge for background subtraction methods. Care must be taken so that they are not 'learned' into the background. Maritime surveillance applications must be able to detect stationary and slow moving objects, as well as fast-moving ones. Also, because of the large viewing ranges, a distant object can appear stationary or slow moving, even if it is actually moving quite quickly. In addition, a common feature of maritime scenes is wake created by boats moving through the water. This appears as a persistent change in the background, so a different form of processing would be required to suppress it.

2.2.2 Foreground segmentation

There are a number of standard segmentation algorithms which have been applied to maritime scenes for the object detection problem. These methods leverage image features (colour, texture, etc.) to divide the image into groups of pixels with related properties which represent higher-level structures in the scene. The fast graph-based segmentation method [74] is used by Socek et al. [202] to segment images from a visual camera based on colour. Bao et al. [15, 16] use the same approach and then label each segment as 'water' or 'non-water' using a SVM classifier trained offline. Villiers et al. [60] use a level set (with Chan-Vese energy minimisation) to segment up to five objects in real time.

Segmentation methods can perform better than background subtraction because they utilise information in the image such as gradients, edges and texture, in addition to pixel colour or intensity. This also means they can exploit certain known properties of objects and backgrounds to distinguish true targets and reduce false positives. However, they rely on good contrast between objects and background regions in order to segment objects accurately. In maritime environments, contrast is often reduced by atmospheric conditions (haze, rain, fog, etc.) or by strong glare from the sun. This is especially the case with distant objects, which makes them less suitable for early piracy detection. Conversely, contrast between wake and sea is usually very high, so these regions are likely to be segmented incorrectly as false positives.

2.2.3 Object models

Another popular approach is to train a classifier to respond to specific characteristics of the objects of interest. Models can be colour-based [205, 206, 232], gradient-based [70–72],

or use more complex features, such as SIFT features [43], Haar classifiers [25, 26, 29] or HOG features [48, 141, 228].

This approach is more robust to dynamic background. However, it is very difficult to design hand-crafted features which are discriminative enough for complex natural images, and methods which learn features need sufficient training data, which can be difficult to obtain in the maritime domain. Restricted training data leads to a limited ability to generalise, both for new object classes and for the same objects viewed under different conditions, viewing angles or scales. Maritime scenes contain diverse lighting conditions, target viewing angles and distances, so this could be problematic. Models which are specific to certain classes of object will not detect other types of object if they happen to enter the scene. If class-agnostic detection is required then multiple detectors must be used.

2.2.4 Deep learning

Deep learning has started to enter the maritime domain in recent years. Popular choices of network are Fast R-CNN [36, 146], Faster R-CNN [50, 116, 210] and Mask R-CNN [149], whilst Cruz et al. [53, 54] use DetectNet[1]. Networks are generally pre-trained on ImageNet [188] and then fine-tuned on a much smaller domain and task-specific dataset. The use of recurrent networks is not common yet in the maritme domain, but [55] is pioneering the use of LSTMs.

Deep learning addresses the limitations of generalising across different scenes and objects, but to do this requires significant amounts of training data. Whilst very large datasets are now available for training deep networks [67, 126, 135, 188], they do not necessarily contain sufficient images which are representative of maritime surveillance scenes. The situation is worse when considering the piracy use case specifically. Fine-tuning on additional data can address this, but labelled data in the public domain is very limited and collecting new data at sea is expensive.

2.3 Saliency, visual attention and salient object detection

One reason why human lookouts remain the state of the art for maritime surveillance is the far superior capabilities of the human visual system compared to cameras. Human

[1] https://devblogs.nvidia.com/detectnet-deep-neural-network-object-detection-digits

eyes have a very large dynamic range and perceive a richer colour spectrum than it is possible to capture with cameras. This gives us a superior ability to correct for lighting, exposure and contrast. In addition, we do not have to have seen a certain type of object before in order to perceive it. These are desirable properties to emulate in a system for maritime object detection.

One of the fundamental properties that drives the human visual system to process a scene is **saliency**. Saliency is the property of being noticeable or important. In images, the saliency of a pixel or region is a combination of local and global uniqueness or contrast. It is context-dependent; there is no absolute measure of saliency. Saliency has been explored in the field of cognitive neuroscience, as well as computer vision. As a result, there are two further terms to define:

- **Visual attention** is a process which directs cognitive resources to the most relevant part of an image or scene. The visual attention mechanism is what guides saccades (eye movements) to most efficiently acquire information. A large body of research [42] has attempted to model or simulate the visual attention mechanism with the aim of predicting how a human would look at an image, in terms of which points would be attended to and in what order.
- **Salient object detection** has also been widely studied in the literature [32]. Here the aim is to extract the salient objects witxh pixel-level accuracy. Salient object detection methods therefore tend to be more data-driven and influenced by image processing and probabilistic techniques from classical computer vision and, more recently, deep learning.

2.3.1 Saliency and visual attention-based approaches in the maritime domain

Outside the main body of saliency and visual attention literature, there are a number of works which have adopted this approach specifically for the maritime object detection task, motivated by the desirable properties described above. In Chapter 4, these saliency-based methods are evaluated on maritime surveillance data to compare their performance and select the most promising for further investigation.

In Table 2.1, the methods are summarised to compare and contrast the different technical approaches used to achieve the visual attention concept. The efforts in this

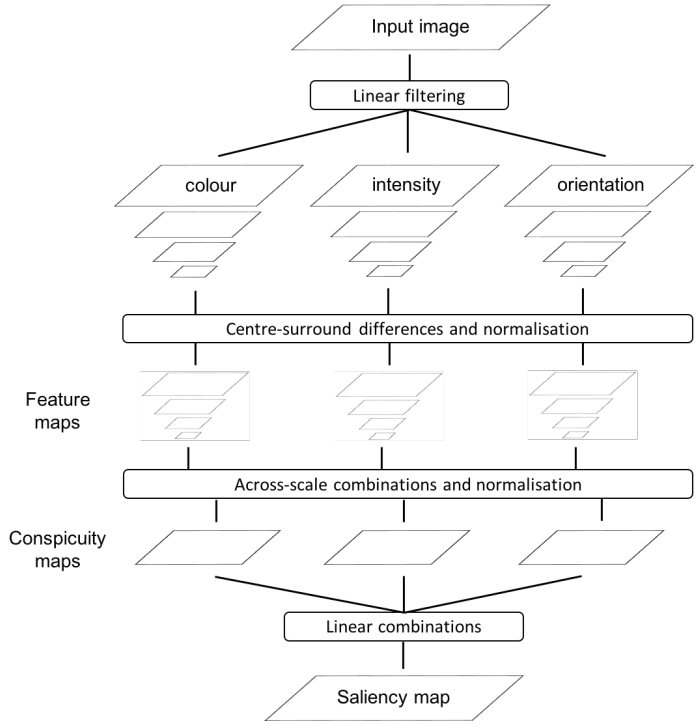


Fig. 2.1 Itti & Koch computational model (adapted from [105])

space can be divided into two main categories: those which have adopted biologically-inspired methods [6, 7, 59, 136, 144, 145, 201] and those which use frequency domain analysis techniques [175–177, 189, 212, 235].

The biologically-inspired approaches mostly follow the Itti and Koch model of visual attention [104, 105] (Fig. 2.1). In this model, the input image is first decomposed into feature maps in different feature channels, such as colour and orientation. A centre-surround process emphasises where a region of the feature map contrasts with its local neighbourhood. This process is performed on a pyramid of images to capture features at different scales. The features are then normalised and combined across scales, and then across channels, to produce a visual attention map which encodes the saliency of each pixel. The methods can be distinguished by their choice of features, how the centre-surround process is modelled, and how the different feature maps are combined. These are summarised in the *Saliency Steps* column of Table 2.1.

Huang et al. [97] proposed a slightly different model called boolean map theory. Under the boolean map model of visual attention, an observer can only access one feature channel at a time in the form of a boolean map. Zhang & Sclaroff [240, 241] formulated the theory into a saliency detection method (Fig. 2.2) called Boolean Map Saliency (BMS). The

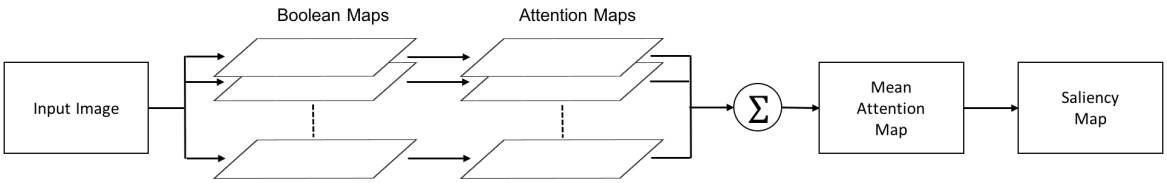


Fig. 2.2 Boolean map saliency block diagram (adapted from [241])

method uses the property of 'surroundedness' to identify regions which receive attention in each boolean map. These are then combined into an overall saliency map.

The frequency domain analysis methods operate mainly in the frequency domain, rather than the spatial domain, and typically stem from signal processing theory and mathematics. These include Principal Components Analysis (PCA), Phase Fourier Transform (PFT), spectral residual (SR) [94] and maximum symmetric surround saliency (MSSS) [3]. The different techniques used are summarised in the *Saliency Steps* column of Table 2.1.

In order to detect objects, most approaches require additional methods to be used in conjunction with the saliency step. This includes background subtraction using a Mixture of Gaussians [144, 212] or learning a background classifier [7], learning weights for combining features in the saliency map [136], and Robust Principal Components Analysis (RPCA) to identify foreground and background from the separated sparse and low-rank matrices [201]. Data-dependent steps are quite common, which means the methods may not generalise well and are difficult to reproduce. The additional processing and data-dependent steps are summarised in Table 2.1.

Table 2.1 Visual attention and saliency based methods in the maritime domain. Key: BI = Biologically-inspired; FD = Frequency Domain Analysis

Approach	Category	Saliency Steps	Additional Steps	Data-dependent Steps
Albrecht [6, 7]	BI	Block-based density, dissimilarity and surround features (edges, colour, right-angles, etc.)	-	Naïve Bayes classifier trained for sea / sky / others based on colour and gradient orientation histograms; Feature integration done through a second Naïve Bayes classifier

Dawkins [59]	BI	Histogram-based colour commonality (7 colour spaces), high pass filtering	Temporal pixel intensity variance and average variance	Binary target / background classifier, linear weights, AdaBoost; Wake suppression with separate classifier
Liu [136]	BI	Local colour and edge orientation histograms comparison across 3 scales	-	Feature integration by learning linear weights for each feature map as per [103]
Makantasis [144, 145]	BI	Edges, horizontal / vertical lines, frequency, colour, entropy – local, global and 'window' levels	MOG background subtraction	Feature integration through SVM binary (target / background) classifier using attention maps and background model features
Sobral [201]	BI	Boolean Map Saliency	RPCA sparse and low rank decomposition background subtraction; Median temporal averaging on BMS	-
Ren 2011 [177]	FD	SVD of intensity channel to get most dominant components, then subtract to leave salient components, then saliency map is further processed by SR and PFT approaches	-	-
Ren 2012 [177]	FD	Spectral residual + spatial filtering on each LAB colour channel	Temporal accumulation of saliency over (e.g. 6) frames	-
Ren 2016 [175]	FD	SVD decomposition of amplitude spectrum on each LAB colour channel, then normalised and combined	-	-
Sadhu [189]	FD	Maximum Symmetric Surround Saliency (MSSS)		Foreground/background linear regression using colour and texture (LBP, entropy) features; Weighted sum of feature regression + MSSS maps
Tran 2016 [212]	FD	Spectral residual and MSSS, 2 maps fused through linear combination with dynamic weights	Background subtraction (MOG, VIBE)	-

Yao 2013 [235]	FD	Intensity image, several scales (fixed image sizes) – spectral residual, followed by MSS on the SR saliency map. Entropy used to select best scale, which is then resized and normalised	-	-

2.3.2 Salient object detection approaches

Given the number of publications in this category, it is surprising that there are no examples which have been applied in the maritime context. There are various reasons why this is perhaps the case. As they are designed for running offline on image sets, rather than live video, they could be too slow for real-time applications. As they have been tailored to compete in saliency benchmarks, they might be biased towards the properties of the datasets used [32]. In particular, many images in saliency benchmarks [33] are photographs where the photographer has pre-selected the salient objects and ensured they are in focus, with balanced colours and lighting, etc. Methods which exploit these factors would not fare well in the real world maritime environment.

For the analysis in Chapter 4, three recent representative methods have been selected to compare against the maritime-specific approaches from Table 2.1. The top performing method (in terms of PR and AUC) – DRFI [222, 223] – and the fastest method – HC [51] – from a recent salient object detection benchmark [33] were selected, along with a more recent method – DSS [92, 93] – which achieves state of the art performance on 5 saliency benchmark datasets. DRFI [222, 223] is similar to the biologically-inspired methods, in that it utilises different features from the image extracted from regions at different scales. A random forest is used to predict the saliency value of each pixel in the image. HC [51] is a histogram contrast-based method which measures saliency of a pixel using colour separation from all other pixels in the image. DSS [92, 93] is a deep-learning based approach which repurposes an earlier deep edge detection network [230] and makes a number of architectural modifications to improve performance.

2.4 Deep learning for semantic segmentation

Semantic segmentation is the process of assigning a class label to every pixel in an image. This is an important task in total scene understanding and is crucial to applications such

as autonomous driving and augmented reality [41, 52, 79]. Deep learning approaches now dominate the field [79]. Driven by the potential application in autonomous vehicle navigation, recent architectures [147, 159] are also specifically designed with speed and memory consumption in mind so that they can run in real time on low-power hardware.

Semantic segmentation has not been used widely in the maritime domain. Kristan et al. [38, 119, 122] have developed a structurally-constrained graphical probabilistic model for segmentation, and objects can be extracted by considering regions which do not fit the model well. The method has been shown to work well, but it requires careful tuning of priors so would need constantly updating for a real-world application. Deep learning-based methods are not common either, as these require large amounts of training data which is not currently available in the maritime domain. However, deep semantic segmentation networks have been used for the purpose of horizon detection [106] and for segmenting ships in remote sensing satellite imagery [224]. A very recent publication [37] conducted similar analysis to that of [45] and Chapter 5 of this thesis but using different networks and different data. Because of its limited use in the maritime domain, semantic segmentation has been chosen as an approach to investigate in more detail.

In Chapter 5, recent semantic segmentation networks are evaluated as the basis for an object detection system. The networks all follow the fully-convolutional encoder-decoder architecture paradigm established by [12] and [140]. The encoder creates a feature hierarchy whilst reducing the spatial resolution. The role of the decoder is then to upsample the low-resolution representation created by the encoder and fine-tune the details to create a pixel-level classification the same size as the original input image. Being fully convolutional reduces the number of network parameters which improves inference speed. It also means they can be applied to input images of any size, irrespective of the size of the training images. This is useful for real-world applications, where the input data may not be the same resolution as the training data.

The networks studied in Chapter 5 all use standard CNN concepts, such as convolution and max-pooling, batch normalisation [101] for accelerating convergence, and dropout [91] for regularisation. In addition, some of the more recent networks use residual connections [90], dilated convolutions [237] and factorised / asymmetric convolutions [9] to improve classification performance and inference speed. The distinctive characteristics of the networks are summarised:

UNet [184] The UNet architecture has a symmetrical encoder-decoder structure such that the output of the first layer can be concatenated with the input to the last layer, the second layer with the second to last, and so on. This is intended to allow the network to make use of high-resolution information from the input when constructing its output.

SegNet [12, 13] SegNet uses the convolutional layers of the VGG16 network [199] as its encoder, and a 'mirror image' of VGG16 as its decoder. The decoder uses pooling indices from the corresponding max-pooling layer of the encoder to create sparsely upsampled feature maps, which are then refined through trainable convolutional filters.

ENet [159] The ENet architecture is based on a 'bottleneck' module, inspired by the residual blocks of ResNet [90]. Dilated convolutions are used in several bottleneck modules to increase the effective receptive field without losing resolution. The decoder is smaller than the encoder to reflect the fact that its main role is to upsample the features, whereas the encoder must learn a good representation in feature space.

ESPNet [147] ESPNet is also based on a repeated module that exploits dilated convolutions and residual connections. Within each ESP module, a spatial pyramid of convolutions learns multi-scale representations simultaneously. Extra efficiency is gained by sometimes using factorised convolutions in place of the normal versions.

ERFNet [182, 183] ERFNet uses a novel redesign of the residual layer in its 'non-bottleneck' configuration, using factorised convolutions to gain speed. Dilated convolutions are also inserted at certain layers to increase the receptive field.

EDANet [139] EDANet is built around the EDA module which exploits asymmetric convolutions for efficiency and dense connections [95] across layers for segmentation accuracy. Like in ERFNet, dilated convolutions are used in some layers to capture more contextual information.

ICNet [245] ICNet uses a cascaded feature fusion process to combine features from different layers of the network and at different resolutions. The input is processed in multiple network branches at different resolutions and the outputs are fused into the final

segmentation. The network is deeply supervised (each branch has its own loss) and speed gains are achieved because the high resolution branch is processed by fewer layers.

As research in this area has progressed, the network architectures have used more sophisticated structures to improve performance, whilst keeping the number of parameters (and therefore inference time) as low as possible. Key advancements, such as the 'bottleneck' module and asymmetric convolutions, reduce computational cost and therefore increase speed. Dilated convolutions are used to increase the effective receptive field of the network, and techniques such as residual connections and combining feature maps from multiple layers improve performance across scales.

From the maritime perspective, the deep semantic segmentation approach is interesting, as the number of images required for training has been observed to be much less (100s of images) [13] than is required for deep object detection networks (millions of images). This makes them attractive for the maritime domain, where there is very little training data available. Furthermore, deep semantic segmentation networks have not been used for object detection in the maritime domain, so they represent an opportunity to investigate a novel approach.

2.5 Maritime object detection in real-world systems

A number of research groups and commercial organisations are developing prototype and operational systems where video analytics techniques are being applied to enhance the surveillance of maritime scenes. Common use cases include vessel traffic monitoring in ports and waterways [25, 168, 179, 195, 221], search and rescue [109, 205, 227, 244], floating object detection [31, 226], collision avoidance [193, 200], and environmental protection [209].

Schwering et al. [194] provide a good overview of what a complete littoral surveillance system must do. They set out the requirements for a programme of research undertaken by the Dutch Ministry of Defence covering requirements for sensors, adaptive sensor management, signal conditioning, and algorithms for detection and classification. Trade-offs are assessed and proposals for potential evaluation criteria are presented. Dijk et al. [61] set out the challenges that are still present seven years later.

In the USA, the US Coast Guard have sponsored a programme to develop the *SeeCoast* system for monitoring ports and harbours [179, 195]. The objective is to reduce the workload for operators monitoring port and harbour regions while maintaining situation awareness. The system performs automated scene understanding with human-in-the-loop control from low-level detection and tracking through to fusion with other sensors (radar, AIS) and activity analysis for alarm generation. Learned normalcy and rule-based anomaly detection are implemented to alert operators.

A slightly different application has been addressed by the *Argos* system which has been put in place along the Grand Canal in Venice for monitoring waterway traffic. A network of cameras along the length of the canal provides a fused situational picture to a central control room. Bloisi et al. [23] report that the system has been running continuously 24 hours a day since 2007 and is able to track boats navigating the channel with good accuracy in real-time.

There are a few companies which offer commercial products and services in this area. *Automatic Sea Vision*[2] (ASV) offer software solutions for automatic detection and tracking of surface targets in a nautical environment [163, 192]. Information is displayed to users in a central display which also allows them to interact with the cameras (e.g. to zoom in or follow detected objects). The system is reported to be used on ships as well as in harbours/ports and for protecting critical infrastructure. A Swedish company, Sjöland & Thyselius, offer a range of services and products for an airborne maritime surveillance system[3] targeted at applications such as oil spill tracking, ship traffic control, search and rescue, fishery surveillance and border control, as well as general surveillance for protection of the Exclusive Economic Zone. The product is a mission management system that presents all available information to the operator in a common situational picture. An integrated ship image database is included in the latest model. The system can be interfaced with visual and IR cameras, as well as radar, AIS and other communications equipment.

The use of vision-based object detection in real world maritime surveillance systems is not widespread, especially in commercial products. Systems and methods that address piracy detection specifically are less common still. This area therefore represents a great opportunity in which to advance the research around vision-based object detection methods and maritime surveillance systems.

[2] www.asv.fr
[3] www.st.se/airborne-systems

2.6 The IPATCH project

The IPATCH project [102] was a collaborative European research project funded by the European 7th Framework Programme. The topic it addressed was "Non-military protection measures for merchant shipping against piracy". The three main goals of the IPATCH project were: 1) to perform an in-depth analysis of the legal, ethical, economic and societal implications of existing counter-piracy measures; 2) to produce well-founded recommendations to the shipping industry to support the use and further development of countermeasures; and 3) to develop a prototype on-board automated surveillance and decision support system which could provide early detection of piracy threats and support the captain and crew in selecting the most appropriate countermeasures.

As part of the project, data collection campaigns were carried out in which cameras were mounted on a vessel at sea and small speedboats acted out realistic piracy scenarios. Four visual and three thermal cameras were used, and the video data was supplemented by recordings of the data feeds from the ship's radar, AIS and navigational systems. The prototype system was also set up and trialled live on two occasions: once on a Naval training vessel in France and once on a crude oil tanker in Greece. This allowed the system to be tested under realistic conditions, as well as providing further opportunities to collect data for algorithm development and performance evaluation.

The work in this thesis was carried out alongside the IPATCH project so it provides background and context across the chapters. In Chapter 3, the IPATCH dataset is described in more detail. In Chapter 4 and 5, the proposed object detection methods are evaluated using video sequences recorded during the campaigns. Finally, in Chapter 6, the proposed methods are evaluated by passing their detections to the IPATCH tracking system to be combined with other recorded data.

2.7 Summary

The development of object detection methods in the maritime domain has followed a similar course to that of the wider object detection field. Background detection approaches are the most popular, despite being susceptible to the problems caused by dynamic background and stationary / slow-moving targets, which are very common in maritime surveillance. Other classical approaches have been explored, but there are still weaknesses and limitations relating to the specific challenges of the maritime environment.

The saliency-based approach has been identified by a number of authors as a good alternative to pursue. Visual saliency is a promising approach for the maritime surveillance case because it should be inherently robust to dynamic background and environmental conditions. It is also class-agnostic, meaning that it can be used for any type of object and in lots of different applications. In Chapter 4, this body of work is analysed thoroughly to compare the methods and identify ways to improve the object detection performance.

The use of deep semantic segmentation networks is under-explored in the maritime domain, so it is interesting to see how this approach performs as the basis for an object detector. Although these networks can learn from smaller training sets, there is still a lack of maritime training data which will need to be overcome through other means.

Finally, there is scope for more research into how vision-based detection methods perform in the context of a wider system. The anti-piracy use case is a valuable case study for this, as it is an important application of this research and has not been widely investigated.

Chapter 3

Evaluation and Benchmarking

3.1 Introduction

Performance evaluation of algorithms allows the objective and quantified comparison of different methods, helps determine the effect of changing different parameters within an algorithm, and can help select the most suitable methods for use in a particular application or under certain conditions. When developing algorithms, benchmarking performance against previous work can confirm that progress is being made in extending the capabilities or addressing weaknesses.

In this chapter, the evaluation and benchmarking methodology that will be used in the experiments of this thesis is laid out, and the baseline methods against which the proposed methods will be compared are described. Related work on performance evaluation from the wider object detection tracking community is summarised, but there are aspects of the maritime surveillance task which require a more tailored approach. Recent maritime-oriented metrics which aim to address this are analysed and further improvements are proposed. Finally, improvements to the Mean Absolute Error metric are proposed to make it invariant to object size when evaluating saliency maps.

3.2 Object detection and tracking

3.2.1 Datasets and challenges

In the wider object detection and tracking community, there have been many workshops, campaigns and projects built around performance evaluation on a curated dataset, often

accompanied by a challenge where participants compete for the best results. Some have focussed on specific classes of object (e.g. pedestrians [63, 222, 225], cyclists [131], faces [14, 190, 225], traffic signs [127] or license plates [231]), whereas others have focussed on detecting or tracking objects in a specific context (e.g. security surveillance [215], protection of mobile assets [161], crowds [75] and autonomous driving [80]). There are also challenges which focus primarily on tracking, such as the Multi Object Tracking (MOT) challenge[1] [129, 148] and the Visual Object Tracking (VOT) challenge[2] [120].

A number of very large datasets have been developed to meet the needs of object detection methods which require large training sets. The PASCAL Visual Object Classes (VOC)[3] [67], ImageNet Large Scale Visual Recognition Challenge (ILSVRC)[4] [188], Common Objects in Context (COCO)[5] [135] and Google Open Images[6] [126] datasets contain millions of images covering thousands of classes and have become the *de facto* standard for training and benchmarking performance.

In this thesis, the task of interest is object detection in maritime surveillance video data. Whilst the above challenges and datasets offer a large amount of high quality data, they do not cover the range of object characteristics and scenes which are representative of the maritime domain. Whilst some of the larger datasets, such as ImageNet and COCO, have potential for use in pre-training deep networks for object detection, they are not suitable for evaluation of maritime surveillance tasks.

3.2.2 Performance evaluation

In this work, the groundtruth available is in the form of bounding boxes and the focus is on metrics which use region overlap as the fundamental building block to align detections to groundtruth. Other methods have been used, such as intercentroidal distance or 'bounding box distance' [196], but region overlap measured with Intersection over Union (IoU) now dominates. In addition, the proposed methods are class agnostic, so classification performance is not evaluated.

True Positives (correctly detected objects), False Positives (detections which do not correspond to a real object) and False Negatives (missed objects) are the primary perfor-

[1] https://motchallenge.net
[2] http://www.votchallenge.net
[3] http://host.robots.ox.ac.uk/pascal/VOC
[4] http://image-net.org/challenges/LSVRC
[5] http://cocodataset.org
[6] https://storage.googleapis.com/openimages/web/index.html

mance indices of interest and there is a range of measures derived from these basic counts (such as F-Score, False Alarm Rate, True Positive Rate, etc.). These require a decision on how to map a 2D bounding box location to a hit/miss binary classification. This is commonly done by selecting an overlap threshold (e.g. 0.5 / 50%), or plotting a curve for a range of thresholds (e.g. Precision-Recall curves). One disadvantage of these measures is that they do not explicitly convey information about how well the object is localised, and using a discrete binary score (hit/miss) for a continuous localisation measure (bounding box position) does not seem a natural fit.

The CLassification of Events, Activities and Relationships (CLEAR) evaluation [19, 204] and the Video Analysis and Content Extraction (VACE) program [114] were instrumental in establishing a set of metrics for visual object detection and tracking which capture more information about the performance. These have been adapted and built on in subsequent work [47, 151] but the high-level ideas are the same. The core metrics are Multiple Object Detection/Tracking Precision and Accuracy (MODA, MODP, MOTA and MOTP).

Since the PASCAL VOC challenge, the evaluation of object detection has been dominated by Average Precision (AP). Average Precision is borrowed from information retrieval [191] and is computed by taking the average of precision values measured at different recall levels. It is initially computed per class and then averaged across all classes to give an overall score called mean AP (mAP). Precision and recall were originally computed for a single overlap threshold (0.5 in PASCAL VOC). Later, in the COCO challenge, AP was computed at a range of values between 0.5 and 0.95 and the mean of these AP values was taken (also called 'mAP'). This was to reward methods which achieved larger overlap with the groundtruth.

AP is not used in this work for two reasons. Firstly, it does not convey information about the localisation accuracy [154] and secondly, the detector needs to output (per-class) ranking to compute it and not all of the methods under consideration in this thesis output scores / rankings.

3.3 Maritime surveillance datasets

Datasets collected from land using cameras on buildings or the shore are common as they are easy to obtain. Data from cameras on small leisure craft is also quite common. Datasets from on-board larger vessels are more difficult and expensive to obtain. Navies

sometimes provide access to this kind of data to research groups, but it is rarely made available in the public domain because of security issues. Aerial datasets are becoming more common with the rise in use (and fall in cost) of UAVs and drones. The rest of this section briefly summarises datasets which address the maritime surveillance task and explains which ones are selected for experiments in this thesis.

3.3.1 Publicly available datasets

Imagery Library for Intelligent Detection Systems (i-LIDS)

Produced by UK Home Office, the i-LIDS dataset [215] is a rare example of a controlled dataset by an agency that can provide independent evaluation and accreditation of algorithm performance. The data concerns detection of people and vehicles for security surveillance and there are a few sequences with a boat on water. Unfortunately, the dataset and evaluation service was discontinued in March 2015.

Maritime Detection Classification and Tracking Database (MarDCT)

The Maritime Detection, Classification, and Tracking (MarDCT) dataset[7] [27] contains videos and images from multiple sources, including fixed, moving, and pan-tilt-zoom cameras, covering various maritime scenarios. Groundtruth is provided for a selection of the sequences, covering the tasks of detection, tracking and classification. Additional groundtruth has been contributed by Prasad et al. [168], including some horizon groundtruth.

A large portion of the dataset is taken from CCTV cameras around the canals of Venice. This data has been captured as part of the ARGOS project [23]. Based on the definition of 'maritime' in this research (see Chapter 2), the Venice data is out of scope for use in training or evaluation. The sequences are low resolution and many suffer from significant compression and other artefacts. MarDCT sequences were therefore not used extensively in this research.

[7] http://www.dis.uniroma1.it/~labrococo/MAR



Fig. 3.1 Example MarDCT images

Maritime Object Detection Dataset (MODD)

MODD[8] contains videos captured by an Unmanned Surface Vehicle (USV) so the data is characterised by a low viewpoint and large camera motion. Version 1 of the dataset [119] contains 12 sequences with groundtruth for objects on the surface of the water and horizon line to evaluate segmentation of the sky, sea and shore regions. Version 2 [38] is similar but provides stereo camera data. MODD is not used extensively in this thesis as it is not representative of the piracy detection use case. However, it is useful for evaluating the ability of the proposed methods to generalise to other applications.



Fig. 3.2 Example MODD images

MARVEL

MARVEL[9] [11] is a very large dataset for training and evaluation of fine-grained classification and instance recognition and image retrieval of vessels. It contains 2 million images of vessels from the *Shipspotting* website[10], where ship and boat enthusiasts are able to upload pictures of vessels when they are spotted in ports and waterways around the world.

[8] https://www.vicos.si/Downloads/MODD
[9] https://github.com/avaapm/marveldataset2016
[10] http://www.shipspotting.com

Data from *ShipSpotting* has been used previously by Albrecht et al. [6, 7] for testing a saliency-based detection method.

MARVEL is not used in this work for several reasons. The images are labelled with the vessel name, IMO number, type (e.g. bulk carrier, trawler) and other vessel particulars but there is no groundtruth for object detection training or evaluation. The images only contain large vessels, so are not suitable for testing detection of small craft like that ones used by pirates. The images are also carefully taken photos where the object is in focus, well framed, in good lighting conditions and so on. This is not representative of the visual challenges which occur in a real-world surveillance system.

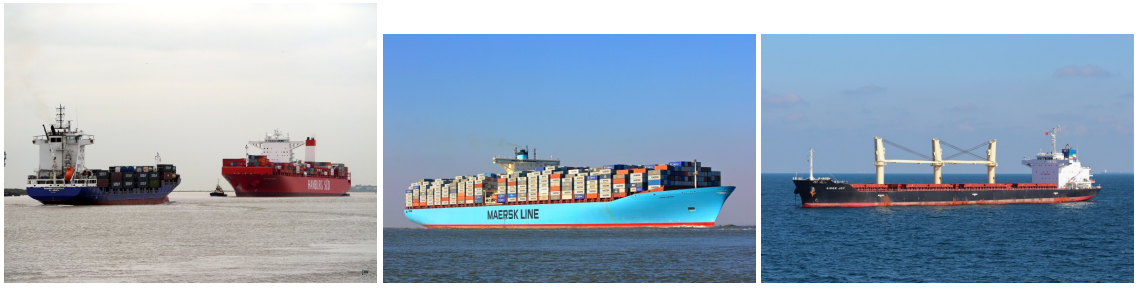


Fig. 3.3 Example MARVEL images

MAritime SATellite Imagery (MASATI)

MASATI[11] is a recently published [77] remote sensing dataset which focusses specifically on ship detection. It provides over 7,000 image tiles taken from Bing maps with accompanying bounding box annotations for vessels. This dataset is out of scope for this work.

PETS 2005

The IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS) has run since 2000, covering a range of themes, challenges and datasets. The theme of the PETS 2005 workshop[12] was target detection and tracking in wide area scenes. One of the datasets made available that year consisted of a Zodiac boat approaching the shore captured by a thermal camera. Fig. 3.5 shows some example frames.

[11] https://www.iuii.ua.es/datasets/masati/index.html
[12] http://www.cvg.reading.ac.uk/PETS2005



Fig. 3.4 Example MASATI images

PETS 2005 is a challenging data set in several respects. In most sequences, the view is not stationary but pans-tilts-zooms to new locations, the images are significantly compressed and noisy, and the objects to be detected are often small and low-contrast. PETS 2005 has been used only a few times in the literature [133, 203, 229]. As there are no visual sequences (only thermal), this dataset is not used in this study.



Fig. 3.5 Example PETS 2005 images

PETS IPATCH

As part of the IPATCH project [102], a large quantity of video data was collected on piracy threat detection scenarios. The sequences consist of small speed boats ('skiffs') approaching a vessel in a variety of patterns to represent different suspicious, threatening and non-suspicious behaviours. The video data is accompanied by AIS and radar data from the host vessel and GPS data from trackers on the skiffs to allow evaluation against groundtruth in real-world coordinates.

Three data collection camapaigns took place during the project in April 2015, September 2016 and May 2017. A portion of the data was first publicly released as part of the PETS

2016 challenge [160]. The following year, the data was re-used in the joint BMTT-PETS workshop [162]. Data from all three campaigns forms the primary basis for evaluation in this thesis and is explained in more detail in Chapter 6.



Fig. 3.6 Example IPATCH images from the 2015, 2016 and 2017 campaigns (left to right)

Singapore Maritime Dataset (SMD)

The Singapore Maritime Dataset (SMD)[13] [168] provides a large set of video sequences captured around Singapore waters from on shore and on board vessels using high resolution visual and NIR cameras. There are 36 sequences comprising more than 17,000 frames. The number of objects per sequence ranges from 2 to 20 and there is a broad range of target sizes due to the large viewing distance. It also provides some more challenging sequences in low light and haze conditions. Groundtruth is provided for evaluation of detection, tracking and horizon detection algorithms. Whilst it does not contain scenes which are representative of the piracy use case, SMD is used in this study because of the complementary challenges it provides alongside the IPATCH dataset.



Fig. 3.7 Example Singapore Maritime Dataset images

SEAGULL

The SEAGULL dataset[14] [180] contains aerial surveillance sequences captured from the point of view of a fixed wing unmanned aerial vehicle (UAV) flying over the sea. A range of

[13] https://sites.google.com/site/dilipprasad/home/singapore-maritime-dataset
[14] http://vislab.isr.ist.utl.pt/seagull-dataset

sensors is used, including visible, NIR, thermal and hyperspectral. Because of the viewpoint, challenges include small targets, significant camera motion and bright reflections. Groundtruth is provided for object detection evaluation. Whilst aerial surveillance is out of scope for evaluation in the IPATCH context, SEAGULL is included in the experiments in this study to see how the proposed methods can generalise to different viewpoints.



Fig. 3.8 Example SEAGULL images

SeaShips

A new dataset designed for training and evaluating ship object detection algorithms has recently been published called SeaShips [197]. The dataset consists of over 31,000 images, covering six vessel types, acquired from the cameras in a coastal video surveillance system. The data covers a range of object scales, hull parts, illumination, viewpoints, backgrounds, and occlusions. Bounding box groundtruth is provided for all images.

Whilst the paper [197] discusses the importance and value of publicly available datasets for benchmarking and performance evaluation, at the time of writing, the data is unfortunately not yet available to download. However, a number of example videos showing object detection results are available as supplementary downloadable material from http://ieeexplore.ieee.org.



Fig. 3.9 Example SeaShips images and detection results (from supplementary material provided with [197])

VAIS

VAIS[15] [242] is a set of images, acquired from thermal and visible cameras simultaneously to support development of multi-modal, fine-grained classification algorithms. It has 1088 pairs of images for 264 unique ships, covering 15 fine-grained categories. The object region is cropped, meaning that many of the images are very small (e.g. 60 × 40 pixels) and would likely present significant challenges to algorithms due to the limited number of features visible. As the focus of this work is object detection, VAIS is not used in the experiments, but some examples of the larger images are shown in Fig. 3.10.



Fig. 3.10 Example VAIS images. Top row: visual camera images, bottom row: corresponding thermal camera images

[15] http://vcipl-okstate.org/pbvs/bench

3.3.2 Non-public datasets

Unfortunately, a lot of papers report results on datasets that are not in the public domain. Often, this is because they have been collected in conjunction with the military (e.g. [146])

SMARTEX

The SMARTEX dataset was collected during the trials of the SMARTEX project in June 2012 in collaboration with the US Coastguard. It consists of sequences from 1 visual and 3 thermal shore-based cameras looking out to sea from a relatively low viewpoint. The objects include fishing boats, sailing boats, cabin cruisers and other small vessels.

The data was kindly provided by colleagues at TNO[16], who were part of the project. The data has not been made public and the work has not been published in a paper, however the authors have requested that a related paper is referenced instead [39, 40].

SMARTEX has not been used extensively in this thesis, as it is primarily thermal imagery and the visual images are low resolution compared to other datasets.

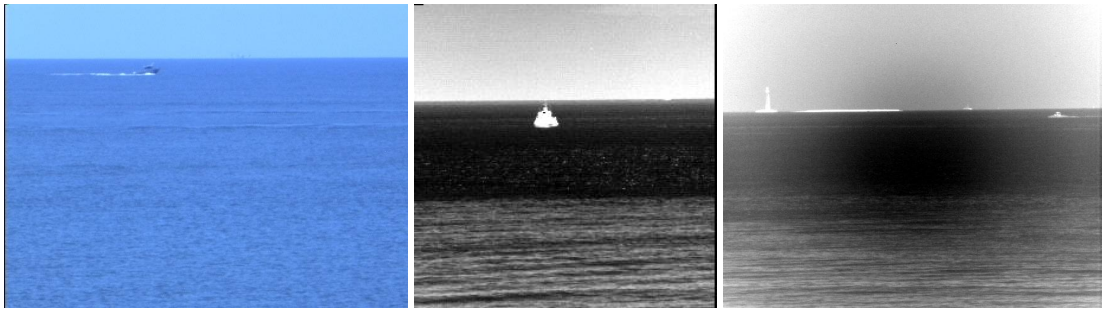


Fig. 3.11 Example SMARTEX images

3.3.3 Synthetic datasets

There are some examples of synthetic data being used but is not common and no examples have been released publicly. For example, [10] created 3D computer models of vessels at different orientations to generate silhouettes. In [78], synthetic images were generated by combining 100 real infrared background images and 80 targets in different locations and sizes. In [102], some synthetic data was created to simulate different wavelengths in thermal cameras (NIR, SWIR, MWIR and LWIR). This was performed using high fidelity

[16]https://www.tno.nl

Table 3.1 Challenges present in maritime datasets

Category	Code	Challenge
-	DB	Dynamic Background *(applicable to all sequences)*
Camera	CM	Camera Motion (Pan / Tilt)
	CZ	Camera Zoom
	LF	Loss of Focus
	EM	Ego Motion of platform
Targets	ST/MT	Single Target, Multiple Targets
	OT	Occluding / Overlapping Targets
	DT	Distant Targets
	SC	Scale Changes
Environment	W	Wake *(caused by motion of object through water))*
	S/R/W	Sparkle / Reflections / Whitepeaks *(caused by natural motions of sea surface interacting with light)*
	DS/G	Direct Sunlight / Glare
	H/R	Haze / Rain
	LL	Low light (e.g. dusk)
Quality	CA	Compression Artefacts
	RA	Recording Artefacts
	V	Vibration

physics models which took high performance computers several days to compute a few minutes of video.

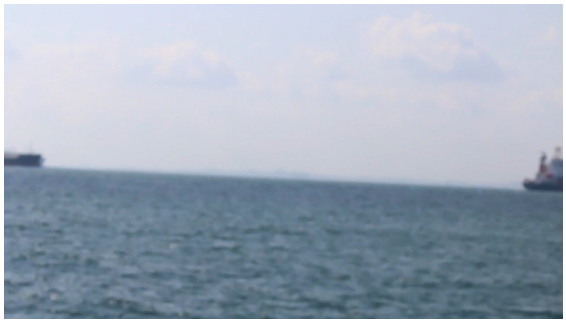
The reason that synthetic data is not common in the literature could be because of the difficulty in simulating the sea in a realistic way. Simulated data is not representative enough of real data to be used for training or testing methods. With the advancements in graphics technology and hardware in the video games industry, this could be addressed in the near future.

3.3.4 Summary and technical challenges

Table 3.1 defines the technical challenges which are present in visual maritime surveillance data. Examples of these are presented in Fig. 3.12. Table 3.2 summarises the dataset properties and technical object detection challenges that are relevant for this research.



(a) Wake in IPATCH



(b) Loss of focus in SMD



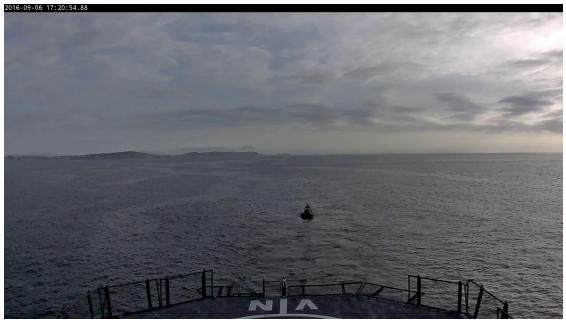
(c) Large reflections in SEAGULL



(d) Sparkle and distant targets in IPATCH



(e) Haze in SMD



(f) Low light in IPATCH

Fig. 3.12 Technical challenges for visual detection in the maritime environment

Table 3.2 Summary of datasets used in this thesis (see Table 3.1 for key to challenges)

							Challenges			
Name	Year	Access	Viewpoints	Modalities	No. sequences	Resolutions	Camera	Targets	Environ.	Quality
MarDCT [27]	2015	Public	Various shore-based	V	28	704 × 576 640 × 360 355 × 288 320 × 240	CM, CZ	ST, MT, OT, DT	W, S/R/W, DS/G	CA, RA
MODD v1 [119]	2016	Public	Very low; camera on USV	V	12	640 × 480	EM	ST, MT, OT, SC	S/R/W, DS/G	-
IPATCH [162]	2016	Public	Medium; camera on large vessel	V, T	113	1920 × 1080	EM	ST, MT, OT, DT, SC	W, S/R/W, DS/G, LL	CA, V
SMD [168]	2017	Public	1) Low; camera on speedboat, 2) Medium; static camera on shore	V, T	36	1920 × 1080	LF, EM	ST, MT, OT, DT	W, S/R/W, H, R, LL	-
SEAGULL [180]	2017	Public	Very high; camera in aerial vehicle	V	75	1920 × 1080 1024 × 768 640 × 480 384 × 288	LF, EM	ST, DT	W, DS/G	V
SMARTEX [39]	2012	Non-public	Low; shore-based	V, T	171	640 × 512 640 × 480 256 × 256	CM, CZ	ST, MT, OT, DT	W	-

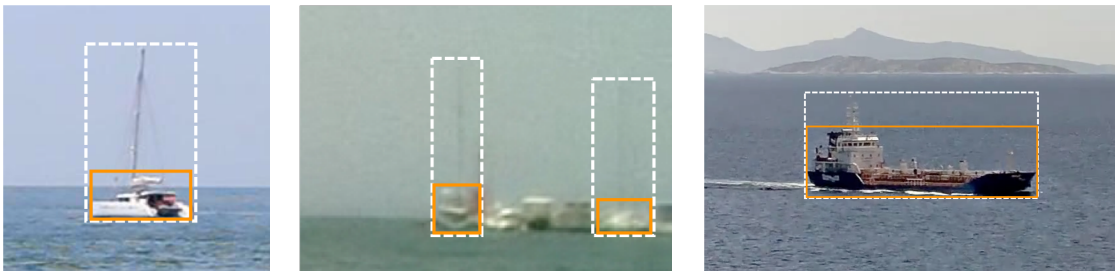
3.4 Metrics for maritime surveillance

The metrics described in Section 3.2.2 are agnostic to any scene context, which makes them good for comparing performance across different tasks and domains but less good for indicating performance in a task or domain-specific way. In visual surveillance, there is usually some context to the object detection task. For example, cameras used in surveillance are often mounted so that they observe the scene roughly horizontally and the right way up. Through calibration, this can be used to convert image coordinates into real-world coordinates. The orientation of the scene can also be used to infer things about the objects, for example that the face of a pedestrian will be in the top half of the bounding box.

In the maritime surveillance context, there are a number of features and trade-offs which are not currently incorporated in standard image-based detection metrics:

- Most boats and vessels have a structure above the main hull which, when viewed from the side, is smaller than the full length. This means that, for a lot of viewing angles, there is a significant portion of empty space in the bounding box region. Small errors in estimating the top of the object can lead to relatively large errors in region overlap. Sailing boats and yachts are more challenging still, as the upper portion consists of masts and ropes which are more difficult to detect, especially at larger distances. Fig. 3.13 shows examples of these characteristics.
- For estimation of the location of an object in the real world (for collision avoidance, piracy warning, etc.), accurate detection of the hull (i.e. the base of the object) is more important that detection of the upper structure (see Fig. 3.14).
- On one hand, the impact of poor detection accuracy is greater at longer distances, due to the non-linear image-depth relationship. On the other hand, there is more time to re-detect and adjust for more distant objects, so closer objects should be more accurately detected.
- For obstacle avoidance and navigation, it is preferable to underestimate distance to the target (appears closer than reality) and overestimate the extent (appears wider than reality).

The first point could be addressed by having multiple bounding boxes or by using pixel-level masks for the groundtruth, but the former could be ambiguous to define and



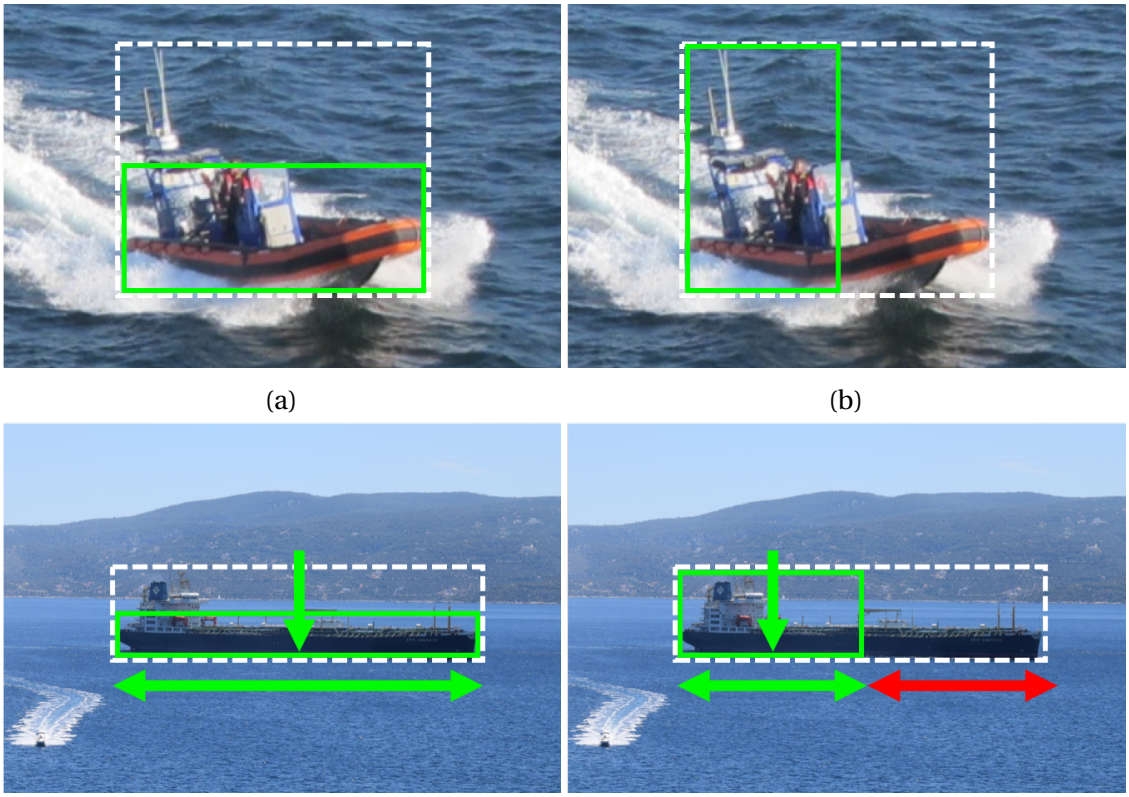
(a) SMD example image (b) MODD example image (c) IPATCH example image

Fig. 3.13 Characteristics of common maritime targets and issues with bounding box-based evaluation. Groundtruth annotations are indicated by the white dotted boxes; many object detectors would output a detection similar to the orange boxes. In (a) and (b), the mast is more difficult to detect as it is narrow and lower contrast than the rest of the boat. However, if it is not detected, a significant area of the region is missed causing a large impact on score. In (c), the antennae on the very top of the vessel are also difficult to detect, but a small inaccuracy in the top edge of the box causes a large error in the area because of the width of the vessel.

evaluate for some objects and the latter is prohibitively time-consuming for the majority of cases.

Instead, it would be better to develop metrics which emphasise the importance of detecting the bottom edge of the bounding box and do not penalise detection of the upper part so strictly. This should not be seen as wanting to design metrics which make algorithms look better on challenging data. The goal is to design metrics which reflect what is important in the task of maritime surveillance to better compare methods, emphasising the relevant aspects for supporting higher level tasks and real-world localisation, not just image-based criteria. With this in mind, the following design criteria are proposed for a maritime-oriented metric:

1. Any portion of the detected region which is outside the groundtruth bounding box should be penalised (includes wake).
2. Errors in width should be penalised equally on either side (symmetric in the x dimension).
3. Errors of the bottom of the box should be penalised more heavily than errors of the top (asymmetric in the y direction).
4. The level of penalisation should change with the size and/or closeness of the object.



(a) (b)

(c) (d)

Fig. 3.14 Importance of detecting hull accurately, compared to upper structures. In the two examples (a-b and c-d), both detections (green boxes) score 50% overlap with the groundtruth (dotted white box). Using MODP or similar, the left cases are indistinguishable from the right cases. From an obstacle detection point of view, the detections in (a) and (c) are preferable, as the location and extent of the vessel has been correctly captured (see green arrows in (c)). In (d), there is a significant error in position and width estimation (red arrow), which could be critical in an obstacle avoidance / navigation system.

5. The metric should yield bounded, meaningful values. Intuitively, a score of zero should indicate a completely missed / failed detection and a score of 1 should indicate a perfect detection (i.e. 100% overlap with groundtruth).

The need for metrics which are more tailored to real-world maritime challenges has also been noted recently by Prasad et al. [164, 167]. They have proposed two measures of precision which emphasise the importance of the bottom edge of the bounding box (see Fig. 3.15 for notation):

$$\text{BEP1} = X_1 Y_1 \tag{3.1a}$$

$$X_1 = \frac{x_b}{x_a + x_b + x_c} \tag{3.1b}$$

$$Y_1 = 1 - \frac{\Delta y_{\text{BE}}}{\min(y_{GT}, y_{DO})} \tag{3.1c}$$

$$\text{BEP2} = X_2 Y_2 \tag{3.2a}$$

$$X_2 = \frac{x_b}{x_a + x_b} \tag{3.2b}$$

$$Y_2 = 1 - \frac{\Delta y_{\text{BE}}}{y_{GT}} \tag{3.2c}$$

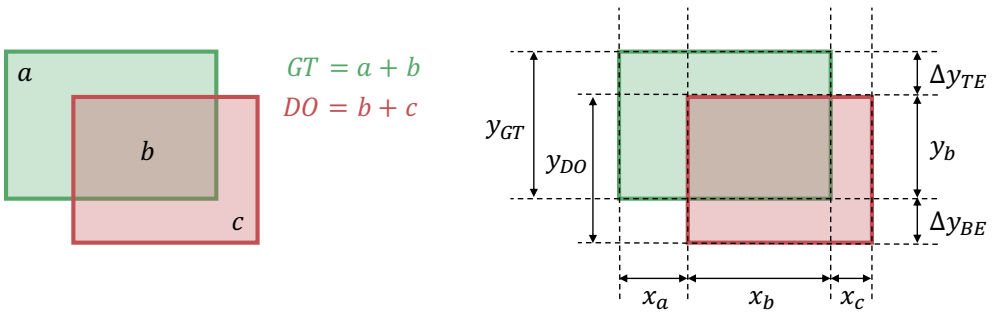
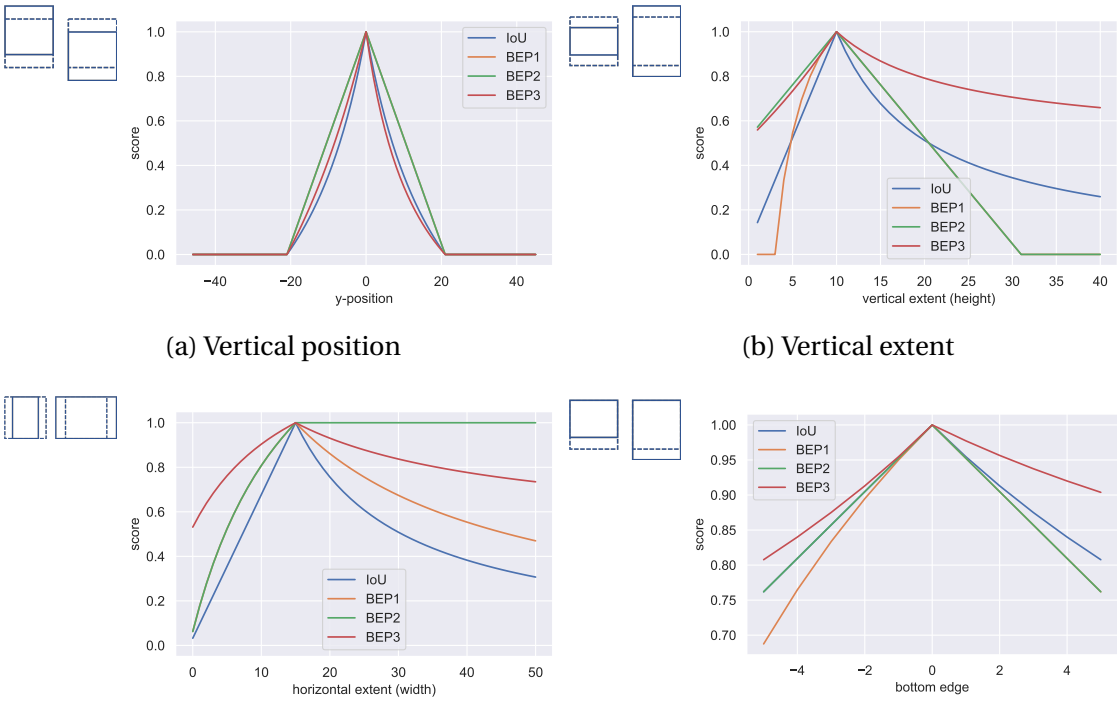


Fig. 3.15 Notation for the bottom-edge precision (BEP) metrics (adapted from [167]). GT = groundtruth, DO = detected object.

Analysis of Prasad metrics



(a) Vertical position

(b) Vertical extent

(c) Horizontal extent

(d) Bottom edge position

Fig. 3.16 Analysis of Prasad BEP metrics (values for BEP3 parameters are $\alpha = 2$, $\beta = 1$ and $\gamma = 0.5$)

Fig. 3.16 shows the response of the Prasad BEP metrics [164, 167] to changes in the detected bounding box (solid line), relative to the groundtruth bounding box (dotted line). Four key dimensions in which the detected box can vary from the groundtruth are analysed: vertical position (y-shift), vertical extent (height), horizontal extent (width) and position of the bottom edge.

BEP2 shows a linear and symmetric response for y-position and bottom edge discrepancy. For horizontal extent, the score saturates when the full groundtruth width is accounted for, regardless of how much it is overestimated. BEP1, on the other hand, penalises overestimations of width, as well as underestimations. In the vertical direction, it increases non-linearly from zero (zero occurs because the base of the detected box is further away than its own height) for underestimates and then decreases linearly (as the size of the groundtruth box is fixed). Like BEP2, BEP1 also has symmetric responses to y-position errors.

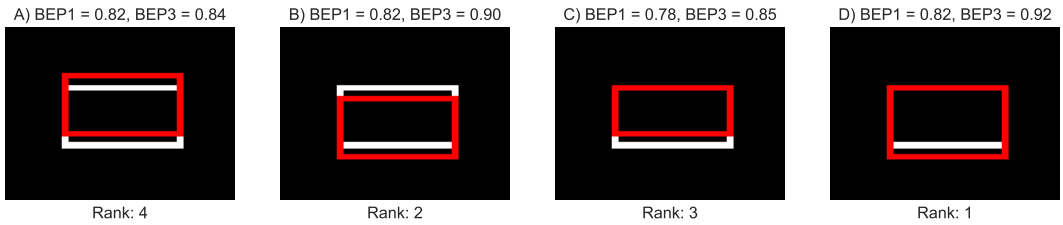


Fig. 3.17 Specific cases and how BEP1 and BEP3 compare to the intuitive ranking (example for $\alpha = 2$, $\beta = 1$, $\gamma = 0.5$)

Whilst these measures are a good step towards creating something more appropriate for maritime, there are still features which are not desirable. The design criteria in [167] relating to obstacle avoidance imply that the metric should be *asymmetric* in the y axis, but in actual fact they are symmetric in both the x and y axes. In addition, the current BEP metrics do not rank detections in the most intuitive order for the maritime case (see Fig. 3.17).

Proposed maritime-oriented metric

Out of the two BEP metrics, BEP1 has more useful properties. The following adaptations to BEP1 are proposed to create BEP3:

$$\text{BEP3} = \begin{cases} 0 & \text{if } X_3 = 0 \text{ or } Y_3 = 0, \\ \frac{X_3 + Y_3}{2} & \text{otherwise} \end{cases} \tag{3.3a}$$

$$X_3 = X_1 = \frac{x_b}{x_a + x_b + x_c} \tag{3.3b}$$

$$Y_3 = \frac{y_b}{y_b + \left(\alpha \Delta y_{BE}^{\text{above}} + \beta \Delta y_{BE}^{\text{below}}\right) + \gamma \Delta y_{TE}} \tag{3.3c}$$

α and β are parameters which penalise bottom edge errors *above* and *below* the groundtruth bottom edge, respectively. γ is a similar parameter for the top edge, but is non-directional (errors above and below are penalised equally). Note that $\Delta y_{BE}^{\text{above}}$ and $\Delta y_{BE}^{\text{below}}$ cannot both be non-zero, so α and β can be tuned independently.

When $\alpha = \beta = \gamma = 0$, the metric does not care about vertical overlap at all and the value (between 0.5 and 1) will only depend on horizontal overlap. When $\alpha = \beta = \gamma = 1$, the metric treats horizontal and vertical overlap equally. For practical applications, users can

choose values which reflect the priorities of their application. In the maritime case, this will typically follow $\alpha > \beta, \gamma =< 1$ to reflect the fact that errors in the bottom edge are more serious if they are above the groundtruth (α) than below (β), and errors in the top edge are less important.

As per criteria 4, it would be useful to be able to take the size and distance of the object into account. This can be done by varying α, β and γ according to some function of the groundtruth size and vertical position in the image. This may be less relevant for aerial surveillance where the viewing angle is very different, in which case the parameters can be set to 1, as above. For all the experiments in this thesis, the values are set to $\alpha = 2, \beta = 1$ and $\gamma = 0.5$.

Note that in BEP_3, the X and Y components are averaged rather than multiplied together. When multiplying, the lower of the two scores dominates (as both components are less that 1). This emphasises the weakest of the two performance dimensions in the combined score and means the metric tends to be concentrated in lower values (i.e. it is a pessimistic metric). By taking the average, the combined score is more balanced and the overall values are more distributed over the range 0 - 1.

Regardless of how the two scores are combined, some information will be lost. In particular, it is difficult to tell the difference between a high and a low score, or two medium scores. However, in this study, the balanced score is selected as it provides a wider range of values, which makes comparison between methods more nuanced. The two components of BEP_3 (X_3 and Y_3) can be analysed individually, as demonstrated in Fig. 3.18 and 3.19.

In Fig. 3.16, the behaviour of the proposed BEP3 measure can be seen alongside IoU, BEP1 and BEP2. BEP3 is now asymmetric with regard to y-position and bottom edge error, with detections above the groundtruth being penalised more heavily. More importantly, this means BEP3 ranks the detections in Fig. 3.17 in the expected order, based on the design criteria for a maritime object detection metric.

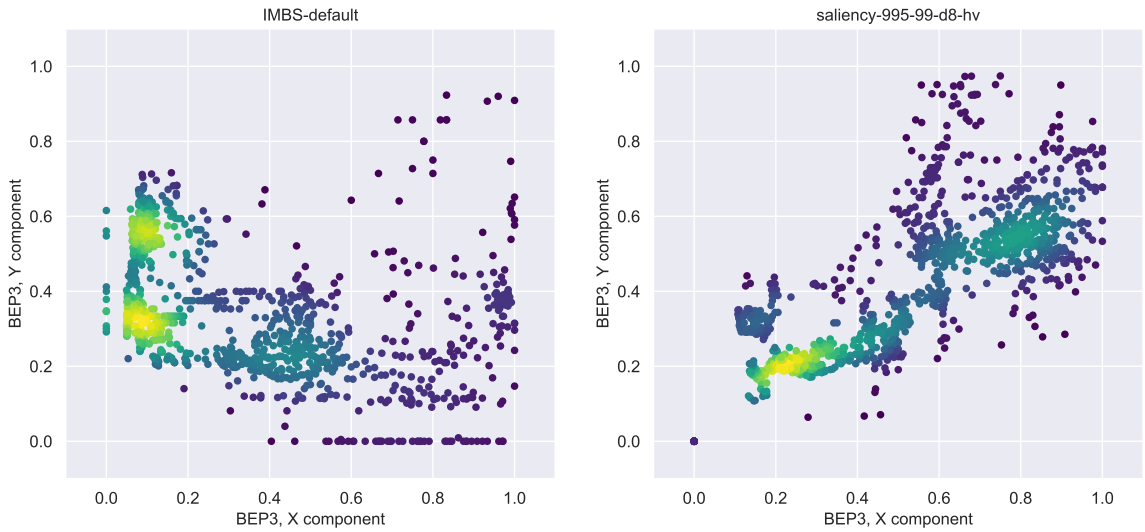


Fig. 3.18 Example showing the X and Y components of BEP3 plotted on a scatter graph ($\alpha = \beta = \gamma = 1$). The points are coloured with a heatmap to show point density within the scatter graph. The IMBS method (left) tends to capture vertical position slightly better than horizontal. This could be because the motion is predominantly horizontally aligned in maritime sequences, so the left and right edges of the bounding box are where background subtraction errors occur. The saliency method (right) has a broader spread of precisions in both dimensions, which reflects the fact its performance is not dependent on motion in the scene.



Fig. 3.19 Example showing the X and Y components of BEP3 plotted against frame number for part of the IPATCH-Sc2a_Tk1-CAM12 ($\alpha = \beta = \gamma = 1$). By looking at the combined score (top), the two methods seem to have similar performance. By looking at the X (middle) and Y (bottom) components, it can be seen that the two methods have opposite strengths (the saliency method is better at capturing the object horizontally but IMBS is better at capturing it vertically). Depending on the application requirements, one method might be preferred over another, which would not have been obvious from the combined BEP3 score.

3.5 Evaluation methodology for maritime object detection in this thesis

This thesis adopts the standard empirical methodology for evaluating computer vision methods, namely comparison of the system output with manually-created groundtruth annotations using a range of metrics. To ensure meaningful results, this methodology requires a broad range of sequences with representative challenges. A set of metrics is used to characterise and quantify different aspects of performance, and the proposed methods are compared against each other and against baseline methods from the literature.

3.5.1 Selection of sequences and groundtruth for object detection evaluation

Table 3.3 lists the sequences which have been selected for object detection performance evaluation. The IPATCH sequences were chosen to represent a range of challenges for the piracy detection use case, including very small/distant objects and appearance changes of approaching targets. The challenge here is to detect the targets as early as possible. To complement the IPATCH data and reflect the ability of the proposed methods to generalise to other scenes and use cases, sequences from SMD and SEAGULL were selected. The SMD sequences contain larger vessels from a stationary shore-based viewpoint and gives the opportunity to test with more than two objects. Two sequences from the SEAGULL dataset are selected to analyse performance on very small objects (10s of pixels) in an aerial surveillance context. All sequences present challenges from the maritime domain, such as wake, sparkle, bright reflections and dynamic background. Groundtruth was used as published for MODD, SMD and SEAGULL, with a few corrections for obvious errors. For IPATCH, groundtruth was created with the ViPER tool [62].

3.5.2 Evaluation metrics used in this thesis

In the literature, it can be ambiguous as to whether a proposed method is a pure detector, a pure tracker, or some combination of both. In both cases, the system outputs an estimated location for all targets in each frame of a video sequence (or image of a dataset). For trackers, there must be a temporal dimension to the task (i.e. operating on a sequence of video frames, compared to pure object detectors which operate on single images).

Table 3.3 Sequences selected for object detection evaluation. Key to challenges in Table 3.1

Dataset	Sequence	Resolution	No. Frames (*with objs.*)	Objs. per Frame		Obj. Size Range (px)		Detection Challenges
				Mean	Max	Width	Height	
IPATCH [162]	2015-Sc2a_Tk1-CAM11	1920 × 1080	3646 (*544*)	1.4	2	5 - 276	8 - 150	EM, MT, SC, W, S/R/W, DS/G, CA
	2015-Sc2a_Tk1-CAM12	1920 × 1080	3857 (*1273*)	1.8	2	3 - 260	8 - 150	EM, MT, OT, DT, SC, W, S/R/W
	2015-Sc3_Tk2-CAM14	1920 × 1080	5425 (*5388*)	2.0	2	7 - 338	7 - 212	EM, MT, DT, SC, W, S/R/W, DS/G
	2016-Sc2_Tk5-CAM11	1920 × 1080	3276 (*3276*)	1.9	2	7 - 82	4 - 77	EM, MT, OT, DT, SC, W, CA, V
	2017-Sc3a-CAM12	1920 × 1080	3028 (*2740*)	2.7	3	7 - 203	7 - 94	EM, MT, DT, SC, W, S/R/W, DS/G, CA
	2017-Sc6b-CAM10	1920 × 1080	3001 (*1855*)	1.6	2	7 - 70	7 - 30	EM, MT, DT, SC, W, S/R/W, DS/G, CA
SMD [168]	1610_VIS_Onshore	1920 × 1080	543 (*543*)	5.8	6	23 - 205	18 - 162	MT, DT, W
	1615_VIS_Onshore	1920 × 1080	566 (*566*)	6.8	8	16 - 533	19 - 203	MT, OT, DT, W
	1619_VIS_Onshore	1920 × 1080	473 (*473*)	6.0	6	66 - 173	46 - 155	MT, DT, W, H/R
SEAGULL [180]	2014-03_bigShipHighAlt clip1	1920 × 1080	1276 (*1276*)	1.0	1	18 - 33	6 - 20	LF, EM, ST, DT, W, S/R/W, DS/G, V
	2014-03_bigShipHighAlt clip2	1920 × 1080	2251 (*1921*)	1.1	2	8 - 40	5 - 32	LF, EM, MT, DT, W, S/R/W, DS/G, V

However, detectors can also use temporal information to detect objects or refine their estimates over time (e.g. background subtraction).

In the context of surveillance, both detection and tracking must be performed by the system. What is needed is a definition of the fundamental tasks that a system must perform as part of detection and tracking. These can then form the basis for evaluation and appropriate measures can be found for them. For object detection, the key questions are:

1. Did it find each object? How many objects did it not find?
2. How good was the spatial localisation (position and extent)?
3. How much was it confused by the background? How many objects did it detect that weren't really objects? (false positives)

Whilst it can be convenient to have a single score with which to compare methods, it is often a better strategy (in terms of interpretation and analysis) to use a complementary set of metrics which measure different aspects of performance. There are usually trade-offs when designing algorithms and fusing all performance aspects into one score can hide these trade-offs. Two algorithms with the same score may have very different precision and accuracy capabilities, for example. It is therefore preferable to use simpler metrics with a well-defined meaning and intuitive relation to the specific task at hand.

In this study, Detection Rate (DR), Multiple Object Detection Precision (MODP) [204] and average False Alarms per Frame (FAF) [148] are selected to meet these criteria. A modified version of MODP is proposed in this section to make it more tailored to the maritime surveillance task, using the BEP3 bottom-edge precision score proposed in Section 3.4. Assignment of detections to groundtruth is performed using the Hungarian algorithm [124, 150].

Metrics are often aggregated over sequences (e.g. averaged over frames). This is acceptable when the target objects do not vary significantly in difficulty (i.e. 'detectability') within a sequence, but averaging can hide details of performance which are useful to know. For example, an object approaching from a large distant will be initially difficult to detect, so it might be informative to see how the detection capability of an algorithm changes as the object gets closer. For this reason, the distributions of metrics are analysed, as well as average (i.e. mean or median) values.

False positives can often be handled in a task-specific way. This means that, when evaluating a detector in relation to a wide range of environments and tasks, the false

positive performance is not the primary concern. When looking at the target use case (piracy), the false positives (quantity and nature) can be analysed *in context*.

Detection Rate (DR)

Detection rate (also known to as true positive rate, sensitivity or recall) is the proportion of groundtruth targets which were successfully detected:

$$\frac{TP}{TP+FN} \tag{3.4}$$

To avoid a single, arbitrary definition of true positive (e.g. 50% IoU) it was decided to compute the value at a range of IoU thresholds and report the results as a curve (see Fig. 3.22 for examples). With low density sequences, such as those in most maritime datasets, the detection rate will take a limited range of values (e.g. 0, 0.5 and 1) for each frame. It is therefore more informative to look at the sequence score (over all frames) rather than frame by frame. The Detection Rate metric as used in this thesis is therefore defined as

$$\widehat{DR}(\tau) = \frac{\sum_t TP^t(\tau)}{\sum_t N_{GT}^t}, \tag{3.5}$$

where TP^t is the number of detections in frame t which exceed an IoU threshold of τ. N_{GT}^t is the number of groundtruth targets in frame t.

Multiple Object Detection Precision (MODP)

The definition of Multiple Object Detection Precision (MODP) from [204] is:

$$\text{MODP}(t) = \frac{\sum_{i=1}^{N_{\text{mapped}}^t} \frac{\left|D_i^t \cap G_i^t\right|}{\left|D_i^t \cup G_i^t\right|}}{N_{\text{mapped}}^t}, \tag{3.6}$$

where

$$\frac{\left|D_i^t \cap G_i^t\right|}{\left|D_i^t \cup G_i^t\right|} \tag{3.7}$$

represents the Intersection over Union overlap ratio between detection D_i and groundtruth G_i in frame t, and N_{mapped}^t is the number of detections which were successfully matched (i.e. IoU > 0) to detections in frame t.

A modified version of MODP is proposed which normalises over the number of groundtruth targets in the frame, rather than the number of mapped targets. This is referred to as MODP-GT:

$$\text{MODP-GT}(t) = \frac{\sum_{i=1}^{N_{\text{GT}}^t} \frac{|D_i^t \cap G_i^t|}{|D_i^t \cup G_i^t|}}{N_{\text{GT}}^t} \tag{3.8}$$

With N_{mapped}^t as the normalisation quantity, MODP tells you, out of the successful detections, what proportion of the target regions was recovered (on average). With N_{GT}^t as the normalising factor, MODP-GT tells you what proportion of the *total possible* combined target region was recovered (on average). It was felt that this interpretation was more intuitive for comparing methods when the number of objects was relatively low, as in the maritime sequences.

The value of MODP-GT is explicitly undefined when there are no groundtruth targets in the frame. This is similar to the original definition of MODP, which states that it should be forced to a value of zero when there are no mapped targets, but this does not allow to distinguish between the case when there are no groundtruth tagets to match and the case where all targets are missed. In the proposed formulation, the former case is undefined and the latter case is zero.

The formulation of MODP makes it easy to incorporate the maritime-specific measures discussed earlier in this chapter. For example, MODP-GT$_{BEP1}$ and MODP-GT$_{BEP3}$ can be created which incorporate the BEP1 and BEP3 precision measures instead of the overlap measure:

$$\text{MODP-GT}_{BEP1}(t) = \frac{\sum_{i=1}^{N_{\text{GT}}^t} \text{BEP1}}{N_{\text{GT}}^t} \tag{3.9}$$

$$\text{MODP-GT}_{BEP3}(t) = \frac{\sum_{i=1}^{N_{\text{GT}}^t} \text{BEP3}}{N_{\text{GT}}^t} \tag{3.10}$$

Fig. 3.20 shows how using these maritime-oriented overlap measures can improve the interpretability of MODP scores, compared to using the standard IoU measure. In particular, Fig. 3.20 shows that MODP-GT$_{BEP3}$ (which uses the proposed BEP3 measure) creates more discriminative results than both MODP-GT (based on IoU) and MODP-GT$_{BEP1}$ (which uses BEP1 [164, 167]).



Fig. 3.20 Example showing effect of BEP3 version compared to MODP (both normalised by total number of groundtruth targets). Two artificial detection outputs were created for the IPATCH-Sc3_Tk2-CAM14 sequence by randomly perturbing the groundtruth bounding box up or down (groundtruth-up and groundtruth-down, respectively). With MODP (top), it is difficult to discriminate between the two outputs. MODP-BEP1 (middle) has the effect of increasing the performance score – which may be misleading – but does not improve the discrimination. The difference becomes clearer with MODP-BEP3 (bottom). Because above-edge errors are penalised more, the groundtruth-down detections get a better MODP-BEP3 score (coefficients used in BEP3: $\alpha = 2$, $\beta = 1$, $\gamma = 0.5$)

As mentioned, the distribution of the MODP-based scores will be analysed, rather than just the mean over a sequence. The scores will be plotted against frame number to show temporal trends and by using boxplots to show distribution characteristics. Examples of this can be seen in Fig. 3.23 and Fig. 3.24. The mean and median values are shown on the boxplots as blue and red bars, respectively. For brevity, MODP-GT$_{BEP3}$ will be referred to as *MODP-BEP3* throughout the rest of this thesis.

False Alarms per Frame (FAF)

False Alarms per Frame (FAF), also known as False Positives Per Image (FPPI) [63], is simply the average number of false positive detections per frame in a sequence:

$$\widehat{FAF} = \frac{\sum_t FP^t}{N_{frames}} \tag{3.11}$$

where FP^t is the number of false positives in frame t and N_{frames} is the number of frames in the sequence. A false positive is defined as any non-matched detection after assignment.

FAF is preferable to False Alarm Rate (FAR) ($= \frac{FP}{FP+TN}$) as it is potentially dominated by the FP term, especially in sequences with only small numbers of objects. Also, when there are no groundtruth targets in a frame, it cannot discriminate between a method which outputs 1 FP and one which outputs 1000 (both = 1), whereas FAF is independent of the number of objects.

3.5.3 Practical upper bound on performance

Bounding box annotations are a good compromise between speed and ease of annotation, and useful results. They are also the most widely used annotation so can be used for comparing with other work. However, when used with metrics that are based on the Intersection over Union / overlap ratio, small differences in bounding box placement can lead to large changes in score. The smaller the object, the bigger the impact this has.

Uncertainty in bounding box placement comes from ambiguity of where the boundary of an object lies. This is more pronounced when target-background contrast is low (e.g. at larger distances or in poor lighting conditions – see Fig. 3.21). There is also naturally some variation in the way different people annotate the same object. In other challenges [114, 121], this variance has been estimated by asking several annotators to annotate the same images and then computing the spread. Only one annotator per sequence

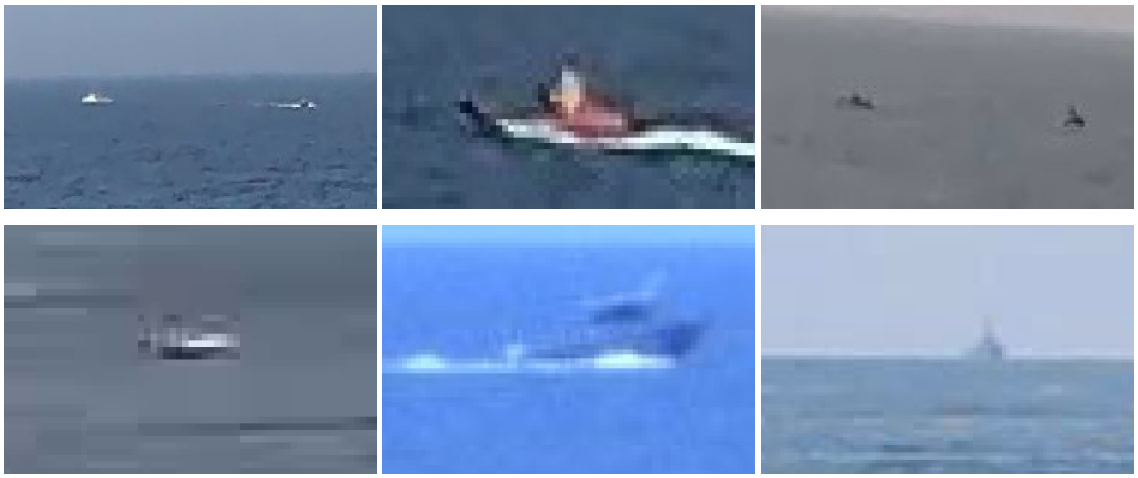


Fig. 3.21 Examples of challenging objects to annotate

was available in this study, so the tolerance of the metrics to small perturbations in the bounding box locations has been estimated in a different way.

For the object detection sequences, the edges of the bounding boxes were randomly perturbed by 1 pixel in every frame and the metrics were computed. The results are shown in Fig. 3.22 to 3.24. This creates a practical upper bound for performance on each sequence, beyond which it is not meaningful to say whether one algorithm is better than another. The 1-pixel perturbation is indicative of the uncertainty introduced by pixel quantisation. The uncertainty created by human annotators may be larger, as differences of more than 1 pixel between annotators are common.

The upper bound depends on the distribution of bounding box sizes in the sequence; a sequence with lots of small boxes will be more influenced by errors. The effect of bounding box size can be seen in Fig. 3.24. At the beginning of the sequence, the targets are very far away and the bounding boxes are very small, so a 1-pixel error causes a larger drop in performance. As the targets approach the camera, the bounding boxes get bigger and the performance improves back to a near-perfect score.

One way of using this information would be to estimate the biggest or average drop in performance for a sequence. This could then be used as a threshold for comparing algorithms on that sequence. If two algorithms have scores within this threshold, they could be considered to have equal performance for that sequence.

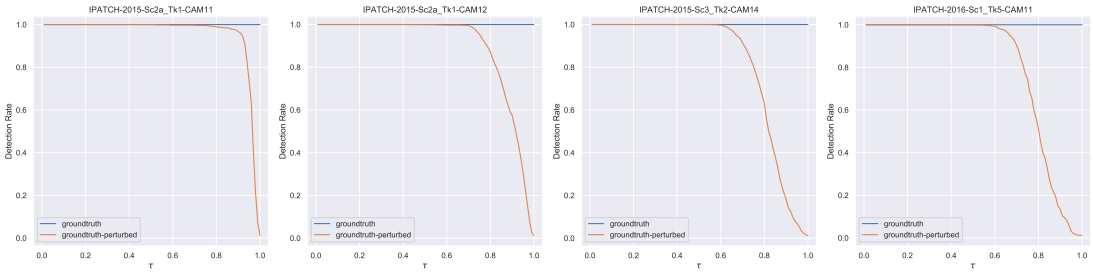


Fig. 3.22 Tolerance of DR curve to 1px perturbations for 4 IPATCH sequences with increasing numbers of small object frames

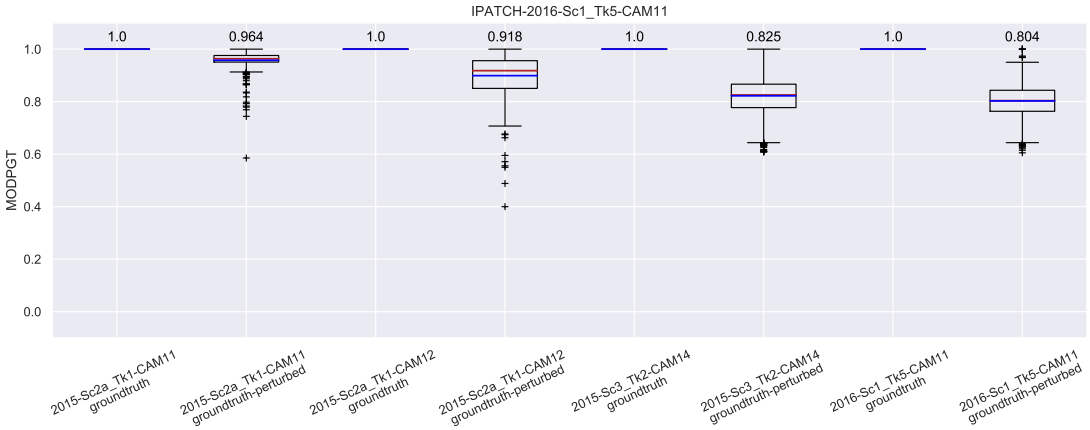


Fig. 3.23 Tolerance of MODP-GT to 1px perturbations for 4 IPATCH sequences with increasing numbers of small object frames

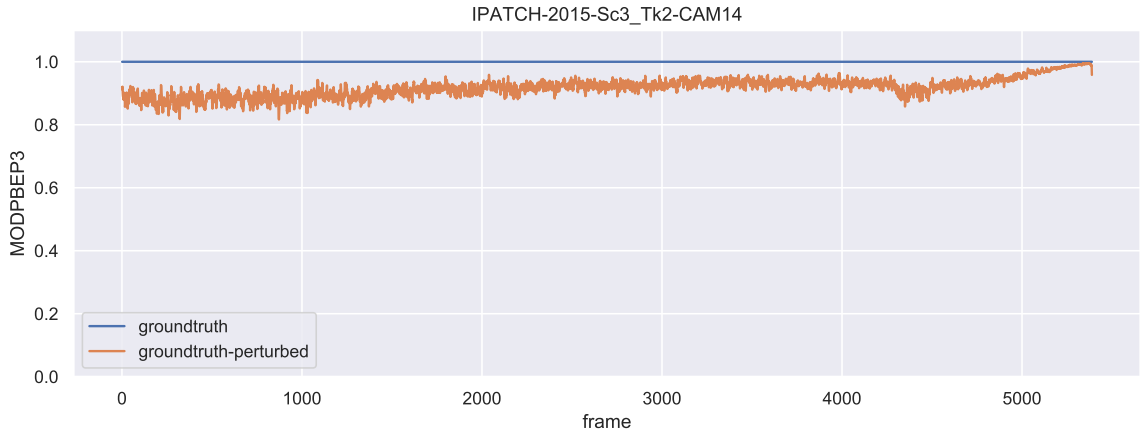


Fig. 3.24 Tolerance of MODP-GT to perturbations with increasing object size within a sequence

3.6 Baseline methods and performance

In order to root the results of this research in existing work, the performance of the proposed object detection methods will be compared against recent maritime-specific methods and current state of the art general object detection approaches. These will be collectively referred to as the 'baselines' and are introduced briefly in this section.

3.6.1 Methods

Independent Multimodal Background Subtraction (IMBS)

There are many background subtraction-based methods in the literature on maritime object detection (see Chapter 2). Independent Multimodal Background Subtraction (IMBS) [24, 28, 30] is chosen as a representative background subtraction method as it is explicitly designed for real-world (and real-time) maritime applications. It uses samples from frames to approximate a discrete distribution for each pixel in the image using online clustering, without assuming a distribution model in advance.

One of the characteristic features of IMBS is a background initialisation phase which is performed in an online, incremental manner to deal with image sequences where no 'clean' frames are available at the beginning of the sequence. However, in the experiments, the online background initialisation process produced many (100,000s) of false positive detections which made evaluation impractical. Therefore, evaluation is not performed on the first 100 frames of each sequence for IMBS to allow time for the background model to establish.

The author's code for the multi-threaded version is used[17]. The sampling period P is set to 160ms and number of samples N is set to 26. This has the effect of triggering a background model refresh every 100 frames. The foreground threshold is set to 50 and all other parameters are kept to the values as reported in [30]. Connected components analysis is used to extract bounding boxes from the binary foreground mask.

Temporally Stable Feature Clusters (TSFC)

The Temporally Stable Feature Cluster (TSFC) method [156] is built on the premise that objects appear as stable structures in the scene, compared to the unstable dynamic

[17] http://users.diag.uniroma1.it/bloisi/sw/imbs-mt.html

background (sea). By tracking features over short periods, the stable features can be extracted. The assumption is that most stable features will be produced by objects, so their locations will be clustered around objects. By clustering the features based on motion and proximity, the bounding boxes of objects can be inferred, whilst transient, uncorrelated motion from the background is suppressed.

The algorithm was implemented in C++ based on the authors' original code with some updates to improve efficiency. The parameters are set as $D_{max} = 0.3$, $\alpha = 5$ and $W_{link} = 10$, with other parameters kept as reported in [156].

Mask R-CNN

Mask R-CNN [89] builds on a family of region proposal networks (R-CNN [83], Fast R-CNN [82] and Faster R-CNN [178]) which exploits a region proposal step to output an arbitrary number of classified bounding boxes per image from a single CNN. Mask R-CNN goes further and uses additional network heads to output instance segmentation masks and person keypoints (left shoulder, right elbow, etc.). For this study, only the bounding box outputs are used. Mask R-CNN is among the current top performing networks in object detection benchmarks, as well as reporting runtime speeds that make it a candidate for use in real-world applications.

The author's code and pre-trained models are used[18]. The models which achieve the fastest processing speed and highest AP score are selected from the *12_2017_baselines* (produced December 2017): *mask_rcnn_R-50-FPN_2x_e2e* (referred to as 'Mask R-CNN R50') and *mask_rcnn_X-101-64x4d-FPN_1x_e2e* (referred to as 'Mask R-CNN X101'). Fine-tuning on maritime data was not performed in order to assess how well the network and model perform 'out of the box'.

YOLO

YOLO [172–174] falls under the category of 'single shot' object detection in which a fixed number of classified bounding boxes are output per image based on a grid cell structure. Single shot methods tend to be faster [96] as there is no region proposal step, and YOLO is one of the fastest object detection networks currently available [174]. However, it does not score as highly in AP as other state of the art networks on benchmark datasets.

[18] https://github.com/facebookresearch/Detectron

The author's code and pre-trained models are used[19]. As with Mask R-CNN, the models with fastest inference speed and highest AP score are selected: YOLO v2-608x608 (referred to as 'YOLO v2') and YOLO v3-608x608 (referred to as 'YOLO v3'). The YOLO v3-tiny model is also included as its speed is an order of magnitude faster than the others. Its AP score is much lower, but it would be interesting to see if this trade-off is worth it. As with Mask R-CNN, fine-tuning was not performed on the YOLO pre-trained models.

3.6.2 Selection and configuration of deep object detection network variants

Table 3.4 Baseline deep object detection variants and reported performance on the COCO dataset [135] (values for Mask R-CNN variants reported in [68], values for YOLO variants reported [174, 171])

Network	Backbone	AP	FPS
Mask R-CNN R50	ResNet-50	37.7	9.9
Mask R-CNN X101	ResNeXt-101	41.3	3.3
YOLO v2	DarkNet-19	21.6	40
YOLO v3	DarkNet-53	33.0	19.6
YOLO v3-tiny	Darknet Reference Model	-	220

The network variants for Mask R-CNN and YOLO are summarised in Table 3.4. For all variants, the backbone network has been pre-trained on the ImageNet 1k dataset [188] and the full network was trained on the COCO 2017 training set [135].

Object detection networks output bounding boxes with a predicted class label and confidence score. As the focus of this study is on class-agnostic object detection, all detections output by the network are used, regardless of their predicted class. The vast majority of detections are predicted as 'boat', but other classes such as 'person' and 'surfboard' are predicted occasionally. This is caused by structures on the targets looking similar to these classes.

Deep object detector networks typically output many bounding boxes clustered around an object location. False detections are removed by setting a minimum confidence score threshold and applying Non-Maximum Suppression (NMS) [73]. This raises the question of which confidence threshold and NMS threshold to select for a given task.

[19] https://github.com/pjreddie/darknet

In challenges such as PASCAL VOC [67], ILSVRC [188] and COCO [135], detectors are evaluated by aggregating their precision scores over a range of recall thresholds to produce the Average Precision (AP) score. This can be further averaged over classes to produce mean Average Precision (mAP). As mentioned previously, AP is not a suitable metric to use for comparison in this work, as the other detection methods do not generate a confidence score or class for each detection.

The 'optimum' confidence and NMS thresholds were therefore derived for each network by minimising an error metric over all the evaluation sequences. This provides a good overall set of parameters for each network which are then used as the baseline for comparison.

Multiple Extended-target Tracking Error (METE) [151] is selected as the metric to optimise as it conveniently combines precision and cardinality errors into a single score. Whilst this might be undesirable in other contexts (c.f. Section 3.5.2), a single value is much easier to optimise. Although METE is defined in the context of tracking, it provides a frame-level measure of performance which is independent of track information (such as track ID), so can be used to evaluate frame-by-frame detection in a video sequence, as is the case here.

METE is computed for each frame, t, and mean METE ($\widehat{\text{METE}}$) is computed over all frames:

$$\text{METE}(t) = \frac{\mathscr{A}_t + \mathscr{C}_t}{max(u_t, v_t)}$$

$$\mathscr{A}_t = \min_{\pi \in \Pi_{max(u_t,v_t)}} \sum_{i=1}^{min(u_t,v_t)} (1 - O(\bar{A}_{t,i}, A_{t,\pi(i)})) \tag{3.12}$$

$$\mathscr{C}_t = |u_t - v_t|$$

$$\widehat{\text{METE}} = \frac{1}{N} \sum_t \text{METE}(t) \tag{3.13}$$

where u_t and v_t are the number of estimated and groundtruth targets, respectively, $O(.)$ is the overlap ratio (IoU), and π is the permutation in the set of permutations Π that maximises the summation term (i.e. the optimal assignment between the estimated targets $\bar{A}_{t,i}$, and the groundtruth, $A_{t,\pi(i)}$). $\mathscr{A}$ and $\mathscr{C}$ represent the accuracy and cardinality errors, respectively.

A set of threshold values $(0.0, 0.1, \ldots, 0.9)$ is selected for confidence[20] and NMS and the network output is evaluated for every combination of these thresholds for each sequence. The METE scores for every frame are computed and averaged over all frames and all sequences. The results are plotted in Figs. 3.26 - 3.28.

Fig. 3.26 shows that, overall, the state of the art deep learning-based object detectors do not achieve particularly good METE scores on the maritime data, regardless of the thresholds used (lower METE is better, as it is an error metric). For the 3 YOLO variants, the choice of NMS threshold makes very little difference as it does not output many detections, even for lower confidence scores.

The nature of the data, in particular the appearance of the objects, plays a big role here. In Fig. 3.27 and 3.28, the same process is repeated, but only taking METE scores from the 3 SMD or 6 IPATCH sequences, respectively. The two datasets contain very different type of object (Fig. 3.25): SMD is primarily large vessels, such as tankers and ferries, whereas IPATCH focusses on small speedboats, similar to those that might be used by pirates.



(a) IPATCH (b) SMD

Fig. 3.25 Examples showing the contrast in the type of target objects from the IPATCH and SMD datasets

The performance of all network variants is much better on the SMD sequences than on the IPATCH ones. It's likely that the boats in the datasets used to train Mask R-CNN and YOLO appear more similar to those in the SMD data than in the IPATCH data. The choice of confidence and NMS threshold has a bigger effect on SMD as the networks are producing more object detections.

In order to compare the Mask R-CNN and YOLO baselines against IMBS, TSFC and the proposed methods, the confidence and NMS thresholds are fixed based on the performance over all sequences. Reading from Fig. 3.26, the smallest (= best) METE score

[20] In practice, the lowest confidence used is 0.05 because the networks output thousands of boxes for any non-zero confidence threshold

is achieved for a confidence threshold of 0.05 and a NMS threshold of 0.1 for both Mask R-CNN variants. For the YOLO variants, the optimal confidence threshold is 0 again, with NMS thresholds of 0.1 for YOLO v2, 0.3 for YOLO v3 and 0.3 for YOLO v3-tiny.

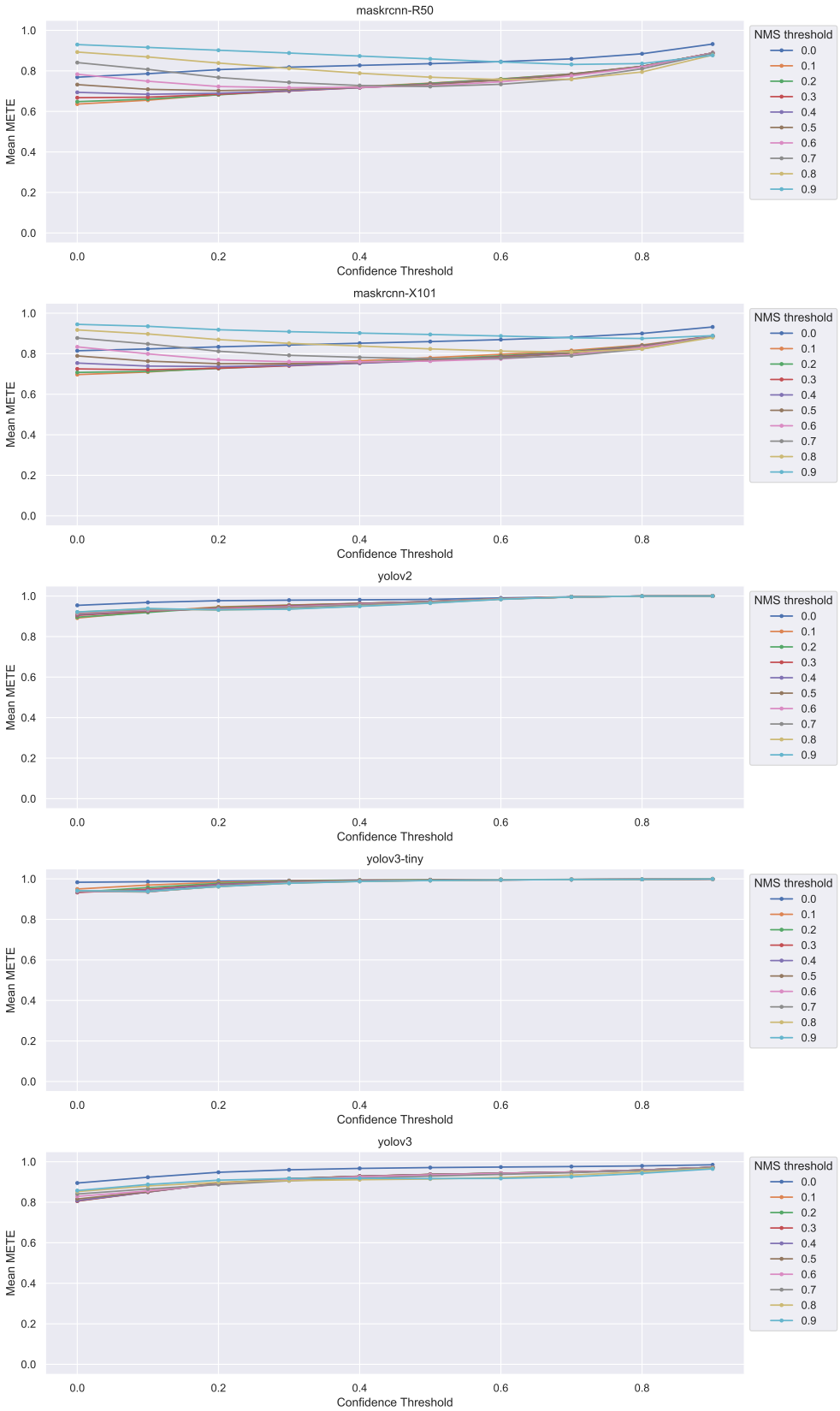


Fig. 3.26 Mean METE for different confidence and NMS thresholds across all sequences

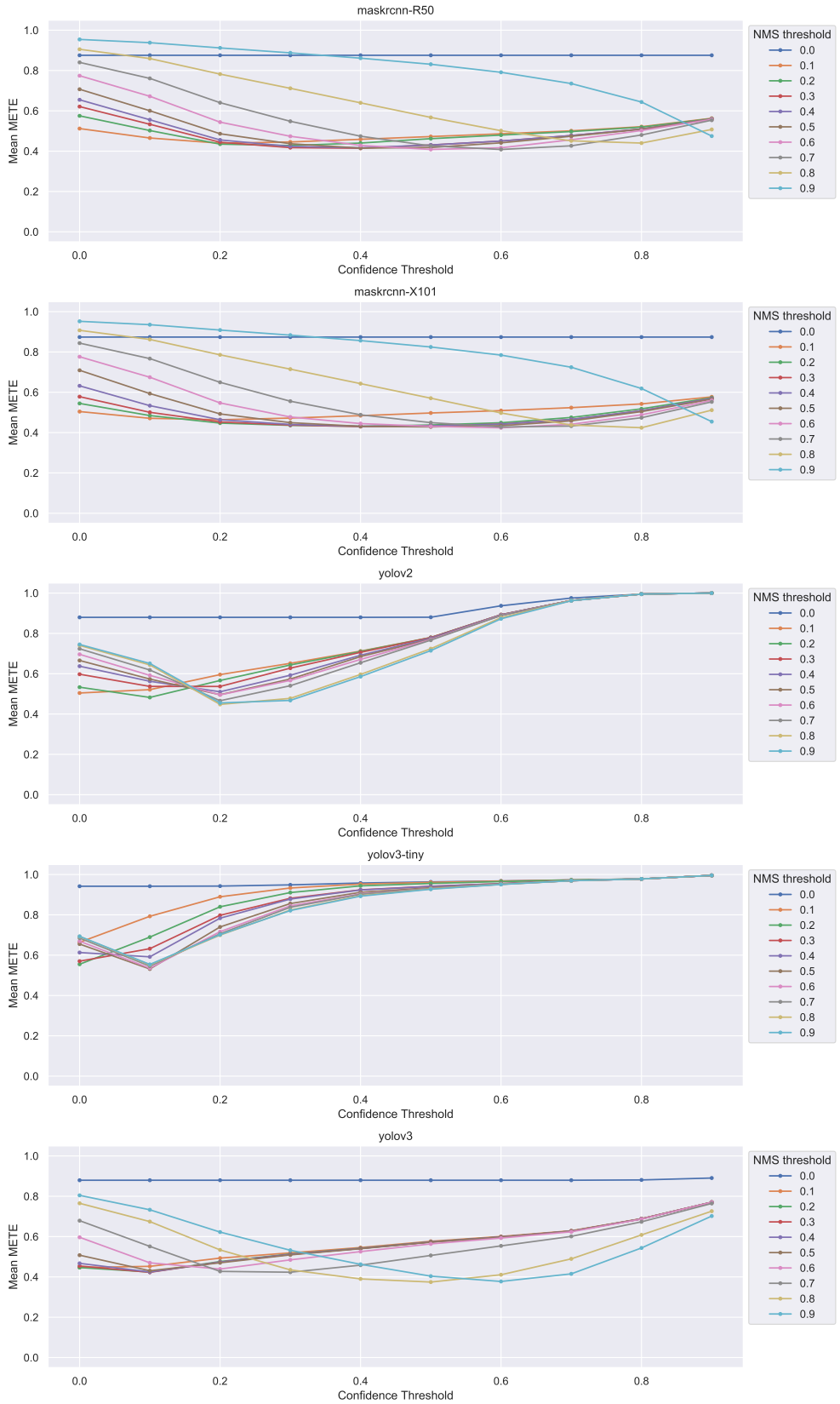


Fig. 3.27 Mean METE for different confidence and NMS thresholds across 3 SMD sequences

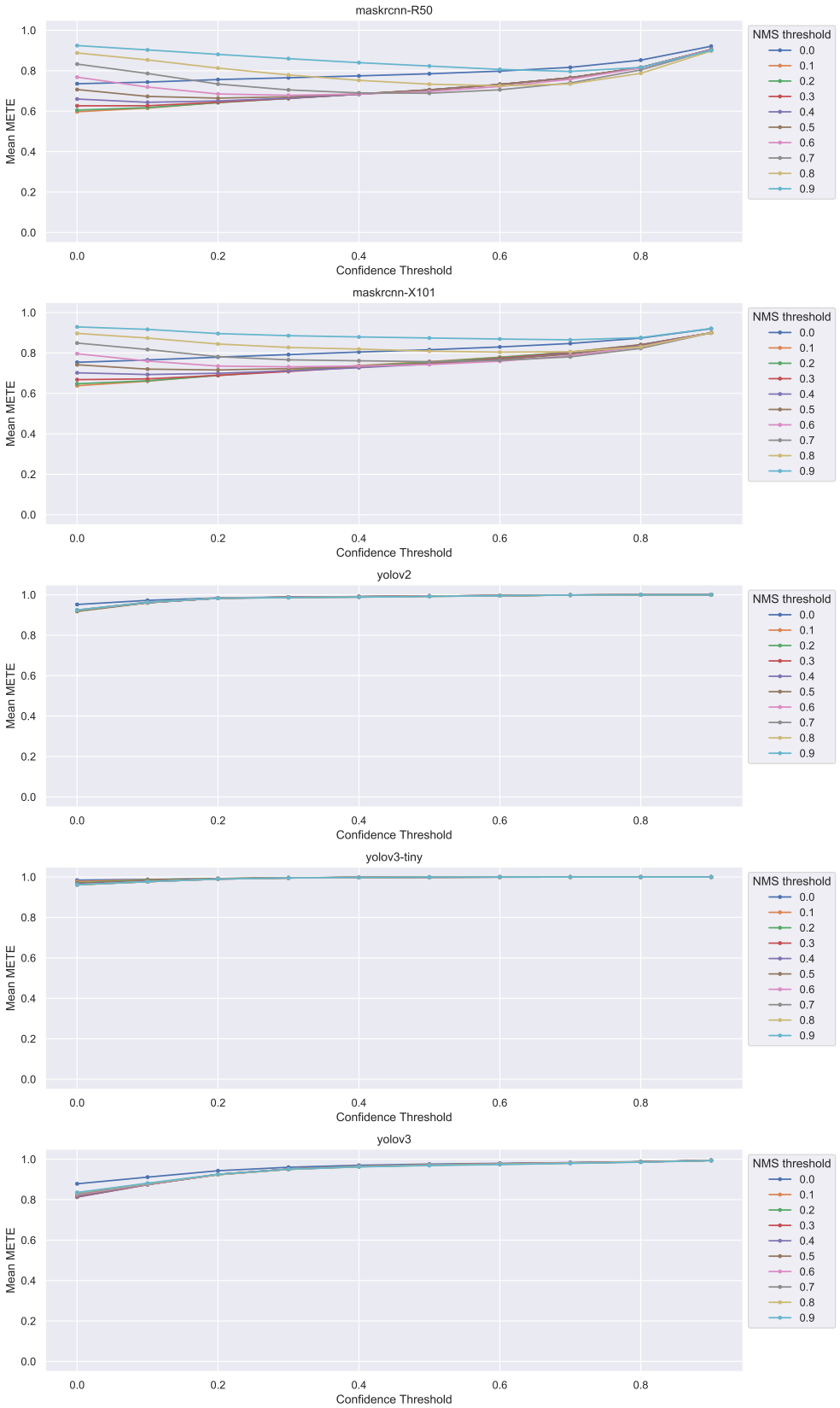


Fig. 3.28 Mean METE for different confidence and NMS thresholds across 4 IPATCH sequences

The network variants were evaluated on the maritime sequences list in Section 3.5.1 according to the methodology outlined in Section 3.5. This was carried out to select the best baseline for each architecture to use in the rest of the experiments. A summary of the MODP-BEP3 results is presented in Table 3.5.

Despite being a more sophisticated architecture, the X101 variant does not perform quite as well as the R50 variant. It is unclear why this is the case. Often, small objects can be missed due to the level of downsampling that occurs within a network. However, looking at the architectures, the level of downsampling is the same in both variants. The two key differences between the R50 (ResNet-50) and X101 (ResNeXt-101) architectures are the additional blocks in the `conv4` layer (6 for R50 vs. 23 for X101) and the structure of each block: ResNet-50 uses standard ResNet bottleneck blocks with a single main path with 64 feature channels, whereas ResNeXt-101 uses blocks with 64 parallel paths of 4 channels each).

Both of these give the X101 architecture has a larger model capacity. This could mean that it is able to learn features which are more specific to each class (it is able to learn more features overall, so it can afford to tailor them more to specific classes). The maritime datasets are generally lower quality than COCO [135], due to low resolution and compression artefacts. Speculating, the highly tuned features that have been learned by the X101 architecture are not 'firing' because of the difference in image characteristics compared to the training data.

Looking at the training documentation and logs[21], the R50 variant was trained for twice as long and with twice the base learning rate as X101 (0.02 vs. 0.01). It's possible that the different training schedules have also contributed to the slightly better performance of the R50 variant. As the R50 variant is also faster than X101 (Fig. 3.29), it is an obvious choice as the Mask R-CNN baseline for further comparisons.

Out of the YOLO variants, YOLO v3 is the only one that is comparable in performance to Mask R-CNN. YOLO v2 struggles with small objects and indeed this was one of the weaknesses addressed in the design of for v3 [174]. YOLO v3 is slightly slower than v2, but this trade-off is worth it. The 'tiny' version of YOLO (yolo-v3-tiny) is very fast, but loses a lot of performance and generates a lot of false positives. The YOLO-v3 network is therefore selected as the YOLO baseline for further comparisons. From this point forward, the two deep network baselines are simply referred to as 'Mask R-CNN' and 'YOLO' for brevity.

[21] https://github.com/facebookresearch/Detectron/blob/master/MODEL_ZOO.md

Table 3.5 Baseline mean and median MODP-BEP3 results for the Mask R-CNN and YOLO variants (Key: M = Mask R-CNN, Y = YOLO, best for each sequence highlighted in green, second best highlighted in blue)

	Median MODP-BEP3					Mean MODP-BEP3				
Sequence	M-R50	M-X101	Y-v2	Y-v3	Y-v3-tiny	M-R50	M-X101	Y-v2	Y-v3	Y-v3-tiny
IPATCH-2015-Sc2a_Tk1-CAM11	**0.772**	**0.778**	0.517	0.743	0.320	**0.730**	**0.745**	0.443	0.668	0.313
IPATCH-2015-Sc2a_Tk1-CAM12	**0.585**	**0.747**	0.238	0.485	0.197	**0.560**	**0.595**	0.253	0.473	0.211
IPATCH-2015-Sc3_Tk2-CAM14	**0.756**	**0.687**	0.000	0.310	0.000	**0.646**	**0.559**	0.082	0.345	0.012
IPATCH-2016-Sc1_Tk5-CAM11	**0.624**	**0.581**	0.000	0.000	0.000	**0.600**	**0.554**	0.101	0.144	0.036
IPATCH-2017-Sc3a-CAM12	0.102	**0.114**	0.035	**0.172**	0.000	**0.190**	**0.245**	0.100	0.180	0.027
IPATCH-2017-Sc6b-CAM10	**0.326**	**0.325**	0.000	0.000	0.000	**0.305**	**0.270**	0.102	0.195	0.053
SMD-Onshore-1610	0.758	**0.764**	0.580	**0.794**	0.281	0.748	**0.761**	0.579	**0.772**	0.282
SMD-Onshore-1615	**0.770**	0.682	0.672	**0.738**	0.330	**0.746**	0.688	0.663	**0.726**	0.336
SMD-Onshore-1619	**0.808**	**0.775**	0.740	0.762	0.473	**0.804**	**0.761**	0.714	0.759	0.469
SEAGULL-bigShipHighAlt_clip1	**0.657**	**0.000**	**0.000**	**0.000**	**0.000**	**0.473**	0.006	0.000	**0.132**	0.000
SEAGULL-bigShipHighAlt_clip2	0.000	0.000	0.000	0.000	0.000	**0.189**	**0.059**	0.015	0.030	0.000

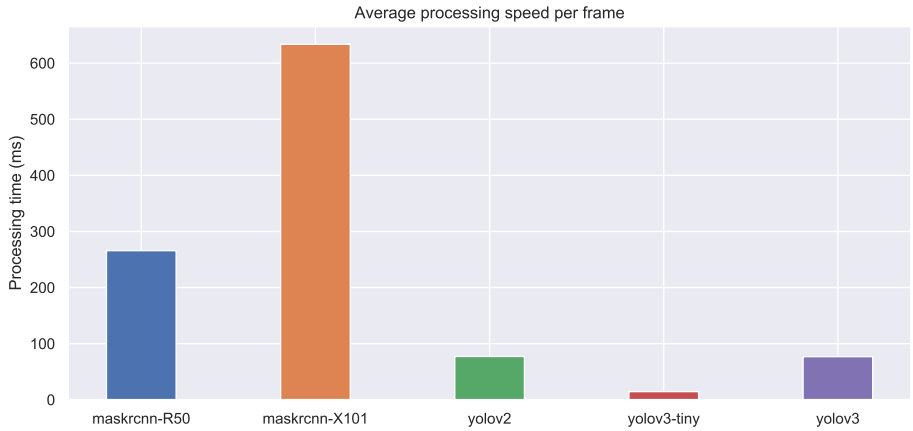


Fig. 3.29 Average processing speed per frame for the deep network variants (benchmarked on an Alienware laptop with an 8-core 2.6GHz Intel® Core™ i7 CPU and 16GB RAM, with an externally connected NVIDIA® GeForce® GTX™ Titan X GPU with 12GB memory)

3.6.3 Performance evaluation

IMBS and TSFC were evaluated on the same sequences as Mask R-CNN R-50 and YOLO v3 and the combined results for the 4 baseline methods are presented in Figs. 3.29 - 3.31 and Table 3.6. This section briefly discusses performance of the baselines, which will be compared against the proposed methods in more detail in subsequent chapters.

Mask R-CNN R-50 has the best overall performance but some sequences challenge its capabilities, in particular IPATCH 2017-Sc3a-CAM12 and SEAGULL bigShipHighAlt_clip2. These sequences both contain distant targets with significant reflections and wake which saturate the white channel. Very distant (i.e. small) targets could be lost in the down-sampling stages of the Mask R-CNN architecture. At the same time, the appearance of the boats is very different to examples seen in the training data. At its best (e.g. SMD-1619-Onshore, Fig. 3.30a), it is able to produce well-localised bounding boxes for every object.

YOLO generally achieves similar performance to Mask R-CNN R-50 but it struggles more with small objects (e.g. IPATCH 2016-Sc1_Tk5-CAM11 and the SEAGULL sequences). This is perhaps due to the fixed-size grid cells which are used to generate the fixed set of bounding box anchors in single-shot detectors, compared to the region proposal networks like Mask R-CNN, where region proposals are generated in a data-driven manner.

A key observation of both the deep learning-based approaches is that they have very low false positive rates (Table 3.6). They tend to either detect an object successfully or output nothing at all. For sequences where they score lower DR and MODP scores, it is because they do not detect some or all of the objects at all, rather than detecting most of the objects with poor localisation. Conversely, because TSFC and IMBS are based on lower-level image features, partial object detections are much more common.

IMBS is often able to locate the targets with good precision but produces a very large number of false positives (Table 3.6). Some could be filtered out based on size, but this would run the risk of accidentally filtering the real targets if they were small (like in IPATCH 2015-Sc3_Tk2 and IPATCH 2016-Sc1_Tk5). More sophisticated filtering (e.g. using classification) could be used but a real system would likely be overwhelmed with so many false positive detections to process. IMBS does particularly well when the sea surface is calm (e.g. IPATCH 2016-Sc1_Tk6) but does not reliably detect stationary objects because they are learned into the background (Fig. 3.30d).

TSFC struggles to detect very small, distant targets (e.g. IPATCH-2015-Sc3_Tk2 and IPATCH-2016-Sc1_Tk5) because it is not able to find enough stable features to form a cluster. It also has a tendency to merge objects which are moving together, such as in IPATCH 2015-Sc2a_Tk1-CAM12 (Fig. 3.30e). Another issue with TSFC is its speed. The processing time grows with the number of features that are being matched from frame to frame, so varies greatly throughout the sequence. A more efficient feature matching mechanism would be needed to make this method suitable for real-time applications.

Finally, wake presents a challenge to all the methods (Fig. 3.30g - 3.30i). TSFC is able to suppress glint and sparkle because their motions are random, but wake contains more stable, structured motion which is incorrectly detected (Fig. 3.30g). IMBS detects wake because it is a significant change to the background distribution of those regions. If wake persists for long enough, it can become part of the new background model. For Mask R-CNN and YOLO, wake is a distractor which creates object-like (specifically, boat-like) features which the network misinterprets as a real object (albeit with low confidence).

Table 3.6 Baseline method FAF

Sequence	Mask R-CNN	YOLO	IMBS	TSFC
IPATCH-2015-Sc2a_Tk1-CAM11	**0.06**	**0.65**	99.99	1.63
IPATCH-2015-Sc2a_Tk1-CAM12	**0.76**	**0.28**	48.09	5.84
IPATCH-2015-Sc3_Tk2-CAM14	**0.15**	**0.08**	118.20	0.55
IPATCH-2016-Sc1_Tk5-CAM11	**0.32**	**0.17**	48.89	1.13
IPATCH-2017-Sc3a-CAM12	**0.02**	**0.07**	168.05	6.05
IPATCH-2017-Sc6b-CAM10	**0.09**	**0.02**	10.64	4.55
SMD-Onshore-1610	1.81	**1.10**	33.50	**0.15**
SMD-Onshore-1615	**2.56**	**1.40**	47.13	3.17
SMD-Onshore-1619	**0.41**	**0.23**	5.70	1.42
SEAGULL-bigShipHighAlt_clip1	**0.00**	**0.00**	5.05	1.43
SEAGULL-bigShipHighAlt_clip2	**0.02**	**0.00**	71.15	7.20



(a) Mask R-CNN R-50, SMD-1619-Onshore (b) YOLO v3, IPATCH 2015-Sc2a_Tk1-CAM11 (c) YOLO v3, SMD-1610-Onshore

(d) IMBS, SMD-1610-Onshore (e) TSFC, IPATCH 2015-Sc2a_Tk1-CAM12 (f) Mask R-CNN R-50, SEAGULL bigShipHighAlt_clip2

(g) TSFC, IPATCH 2017-Sc3a-CAM12 (h) IMBS, IPATCH 2017-Sc6b-CAM10 (i) YOLO v3, IPATCH 2016-Sc1_Tk5-CAM11

Fig. 3.30 Qualitative results for the baseline methods

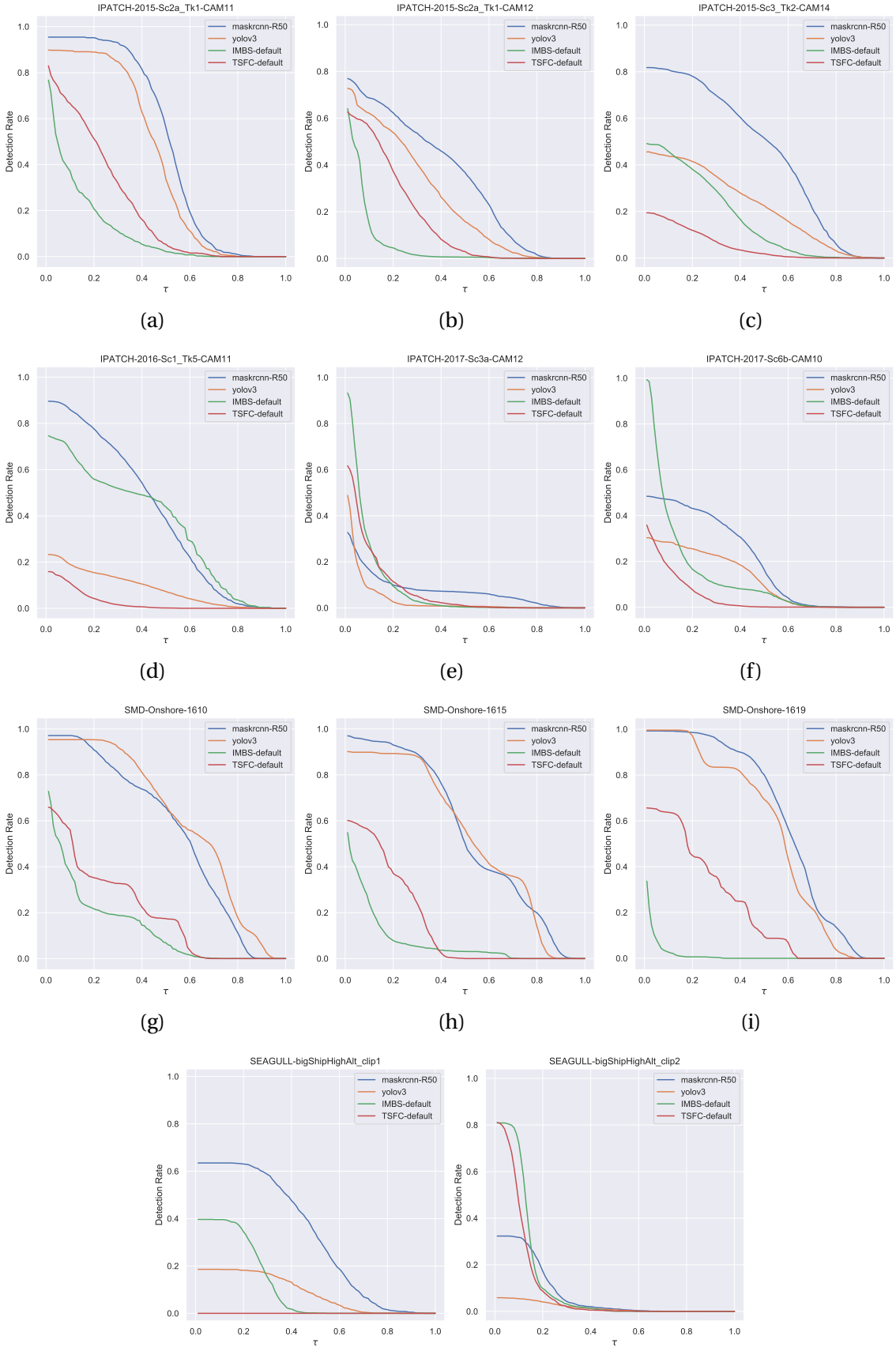
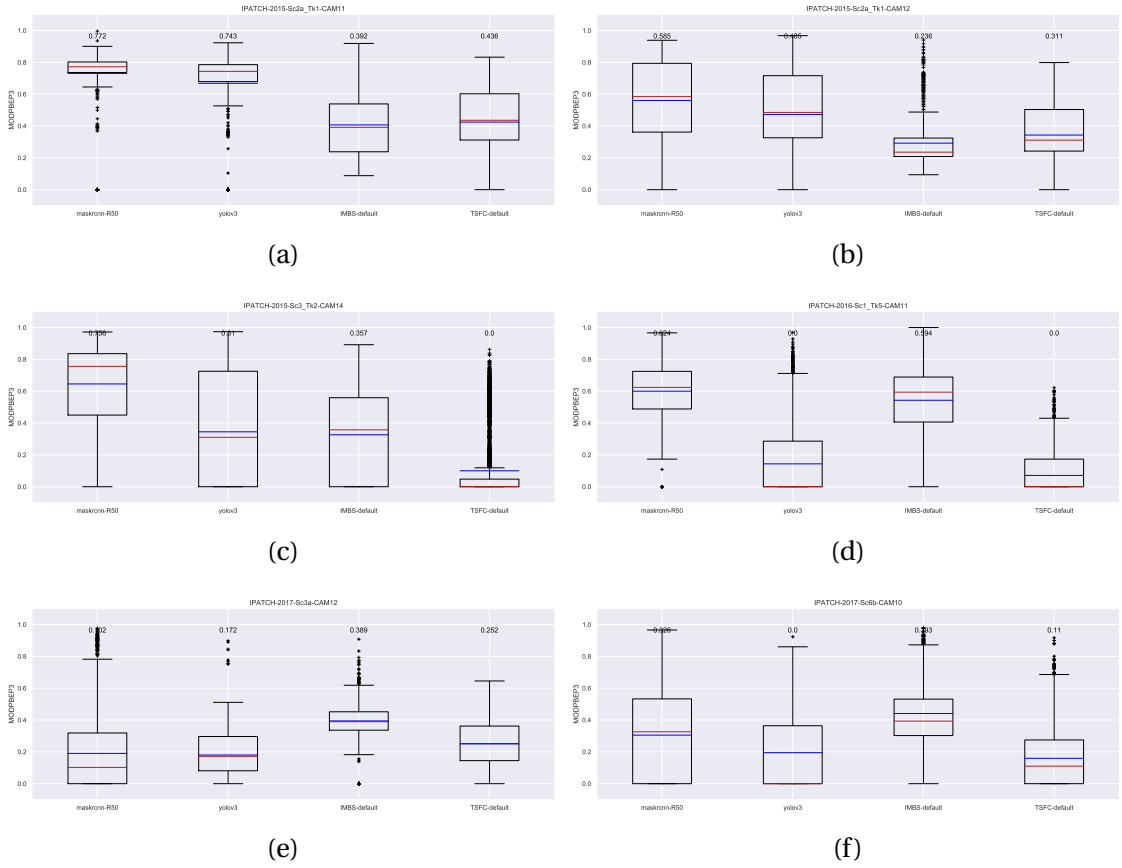
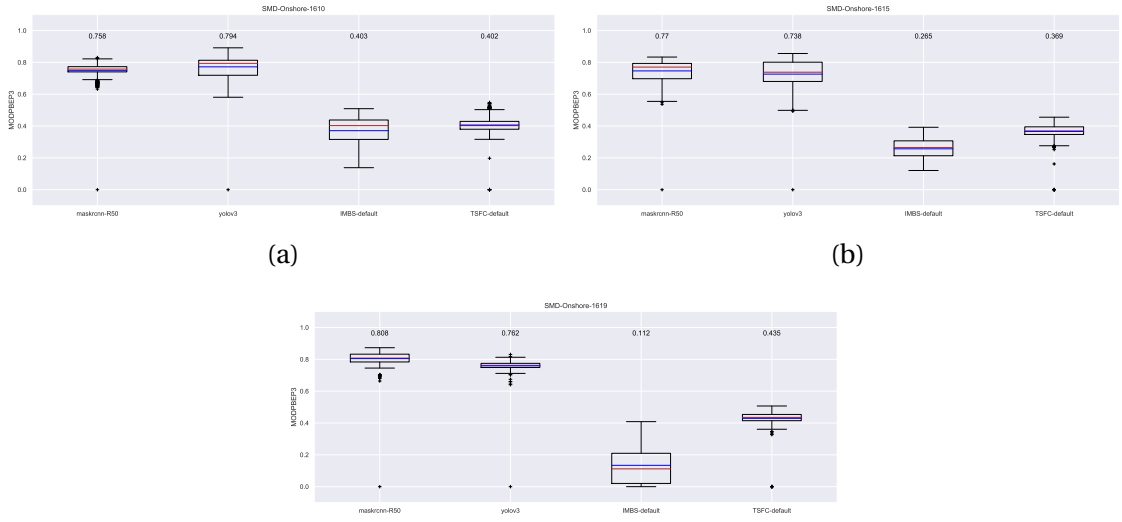


Fig. 3.31 Detection rate curves for the baseline methods



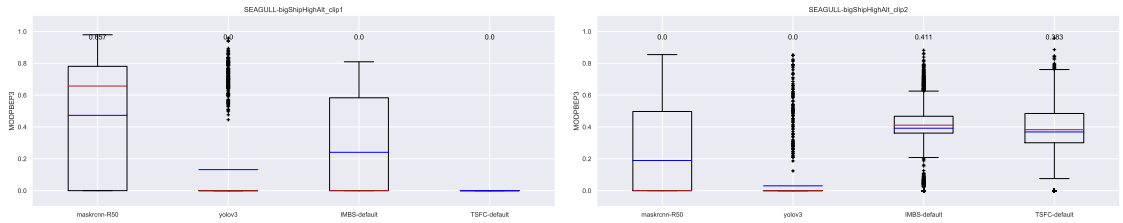
(a) (b) (c) (d) (e) (f)

Fig. 3.32 MODP-BEP3 results for the baseline methods on IPATCH sequences



(a) (b) (c)

Fig. 3.33 MODP-BEP3 results for the baseline methods on SMD sequences



(a) (b)

Fig. 3.34 MODP-BEP3 results for the baseline methods on SEAGULL sequences

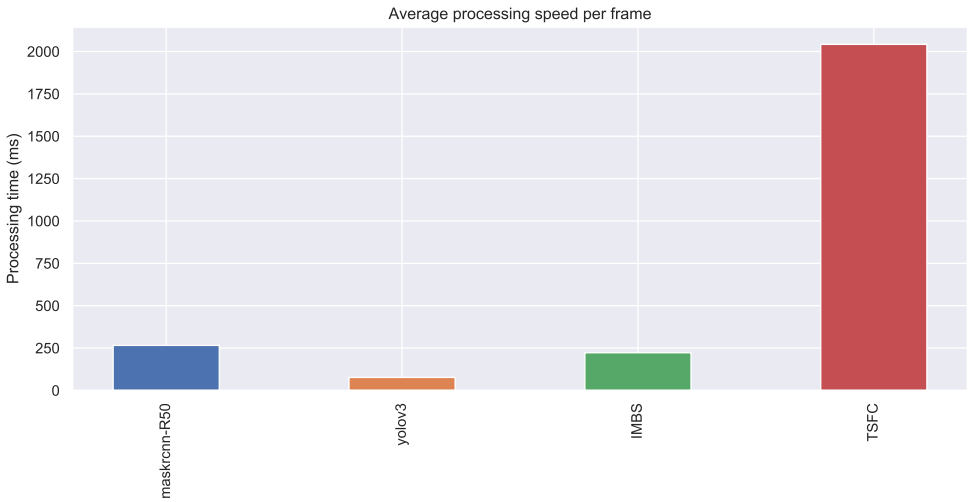
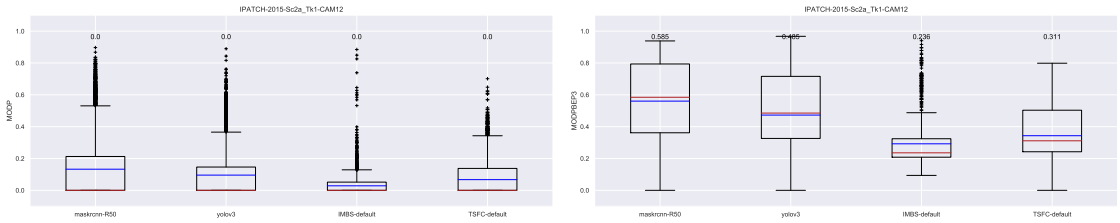


Fig. 3.35 Average processing speed per frame for the baseline methods



(a) Results using MODP (as defined in [204])

(b) Results using MODP-GT$_{BEP3}$

Fig. 3.36 Effect of using MODP-BEP3 instead of MODP for evaluation

3.6.4 MODP vs. MODP-GT$_{BEP3}$

In Section 3.5.2, MODP-GT$_{BEP3}$ was proposed as a more appropriate metric for evaluation maritime object detection. In Fig. 3.36, an example is presented to compare the two metrics side by side. Using MODP (Fig. 3.36a), the median scores are all 0 and the distributions are very similar so it is difficult to compare performance across the methods. With MODP-GT$_{BEP3}$ (Fig. 3.36b), the distributions cover a wider range and features are revealed which were not visible with MODP. For example, the range of MODP-GT$_{BEP3}$ values achieved by the IMBS method does not extend as far as zero in Fig. 3.36b, whereas the other methods do. This detail is lost in Fig. 3.36a where all the MODP values are compressed into a small range.

3.7 Improved Mean Absolute Error for evaluating saliency maps

In Chapter 4, an object detection method is proposed which uses visual saliency. An analysis of different saliency methods is performed to determine which would be the most promising for use in this way. A metric called Mean Absolute Error is used as part of this analysis. This section identifies a possible weakness of this metric and proposes an improved version which will be used in Chapter 4.

3.7.1 Mean Absolute Error (MAE)

Mean Absolute Error (MAE) measures the average deviation between a saliency map and the groundtruth object regions. It is therefore an indication of how well the saliency map models the saliency of the scene. The MAE score for frame n is computed as the average absolute pixel-wise difference between the saliency map, S, and the binary groundtruth mask, G, both scaled to the range $[0,1]$

$$\mathrm{MAE}_n = \frac{1}{W \times H} \sum_{i=1}^{W} \sum_{j=1}^{H} \left| S_n(i,j) - G_n(i,j) \right|, \tag{3.14}$$

where $S_n(i,j)$ and $G_n(i,j)$ are the saliency and groundtruth values of pixel (i,j) in frame n, and W and H are the image width and height. In addition, the mean MAE score, $\widehat{\mathrm{MAE}}$, is calculated for a sequence by averaging over all frames.

$$\widehat{\mathrm{MAE}} = \frac{1}{N} \sum_{n=1}^{N} \mathrm{MAE}_n, \tag{3.15}$$

where N is the number of frames in the sequence.

3.7.2 Shortcomings

It is proposed that a weakness of MAE is that it is dependent on object size within the image. There is an implicit assumption that positive and negative classes in the groundtruth map are approximately equally likely. However, if the object area is small, saliency maps which predict generally low saliency values will score well, even if they do not locate the true salient pixels particularly well. The case is reversed for large salient objects.

For example, consider the case of an object that is 10 × 10 pixels inside a 100 × 100 pixel image. A saliency method that predicts maximum saliency (= 1) everywhere in the map ('all white') will get an MAE of 0.01:

$$\frac{(1 \times 100) + (0 \times 9,900)}{10,000} = 0.01 \tag{3.16}$$

A saliency method that predicts minimum saliency (= 0) everywhere in the map ('all black') will get an MAE of 0.99:

$$\frac{(0 \times 100) + (1 \times 9,900)}{10,000} = 0.99 \tag{3.17}$$

The 'all black' approach achieves a near perfect score, even though it detects none of the salient object pixels.

This effect can be seen if the MAE scores are plotted for 4 different cases: 'all black' (saliency map is 0 everywhere), 'all white' (saliency map is 1 everywhere), 'random' (saliency map sampled from $\mathscr{U}(0,1)$) and 'groundtruth' (i.e. the groundtruth compared with itself). See Fig. 3.37.

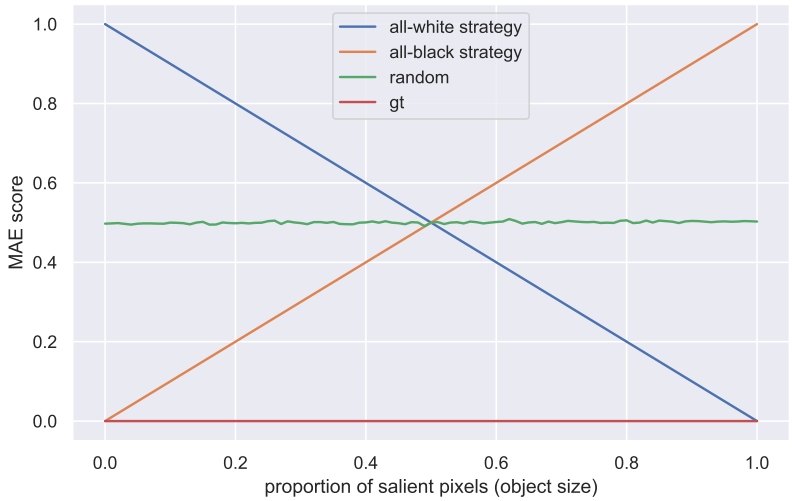


Fig. 3.37 MAE behaviour for different strategies as a function of object size

As expected, the 'random' approach produces an MAE of 0.5 and 'groundtruth' achieves a perfect score of 0 for all object sizes. However, the success of the 'all white' and 'all black' strategies are dependent on the proportion of positive examples in the groundtruth map. The MAE score favours methods which output saliency maps with certain distributions.

This bias in the MAE metric has not been described in the recent salient object detection literature. It's possible that with current saliency benchmarks [33], the bias does not have a significant impact because there is a range of object sizes. However, in the maritime surveillance data used in this work, the objects are all very small compared to the image size, so the imbalance has more of an impact.

3.7.3 Proposed improvements

The new version of the metric should have the following properties:

1. Be bounded (i.e. have a defined range)
2. Yield values with a natural interpretation
3. Be invariant to object size within the image
4. Represent the performance of a saliency method in terms of both false positives and false negatives

A new, balanced version of MAE is proposed which does not suffer from the bias described above. The balanced metric is made up of two components:

$$\mathrm{MAE}^{+} = \frac{1}{N^{+}} \sum_{i} \sum_{j} \left| S^{+}(i,j) - G^{+}(i,j) \right| \tag{3.18}$$

$$\mathrm{MAE}^{-} = \frac{1}{N^{-}} \sum_{i} \sum_{j} \left| S^{-}(i,j) - G^{-}(i,j) \right| \tag{3.19}$$

where G^{+} and G^{-} represents the positive and negative regions of the groundtruth map, respectively. S^{+} and S^{-} are the regions of the saliency map corresponding to the positive and negative regions of the groundtruth map. N^{+} and N^{-} are the number of pixels in the positive and negative groundtruth regions. In other words, MAE^{+} and MAE^{-} are the MAE scores calculated separately for the positive and negative regions of the groundtruth map.

These two components can be combined to give a single error score which balances the two components fairly and also takes into account empty groundtruth maps. The new metric is named "Balanced Mean Absolute Error" or BMAE:

$$\mathrm{BMAE} = \alpha^{\min(N^{-},1)} \mathrm{MAE}^{+} + (1-\alpha)^{\min(N^{+},1)} \mathrm{MAE}^{-} \tag{3.20}$$

The $\alpha^{\min(N,1)}$ structure is a mechanism to handle the edge cases when the groundtruth map is either all zero (no salient objects) or all one. The latter case is very unlikely to happen in maritime surveillance (and most other) scenarios, but the former is very common. The metric can be interpreted as follows:

- A value of 0 indicates a perfect score – the groundtruth has been matched exactly
- A value of 1 is the worst case – the saliency map has predicted the exact opposite of the groundtruth
- Random guessing (e.g. uniformly distributed noise) equates to a score of 0.5
- Most saliency methods will achieve a score between 0 and 0.5, i.e. should be better than random guessing but unlikely to be perfect

Fig. 3.38 shows how the balanced version of MAE is invariant to object size under different strategies.



Fig. 3.38 Behaviour of the proposed Balanced MAE metric under different strategies

The balanced version of MAE proposed in this chapter is used in Chapter 4 to compare the performance of various saliency methods on maritime surveillance data. This is particularly important because maritime surveillance sequences contain small targets (10s of pixels). By using the proposed MAE, algorithms which generally predict higher saliency values are not disproportionately penalised.

3.8 Summary

In this chapter, the evaluation methodology that will be used in the subsequent chapters for analysing the performance of the different object detection methods has been presented. As part of that methodology, a new maritime-oriented metric (MODP-GT_{BEP3}) was proposed, based on previous work in [167], which emphasises the importance of the bottom edge of the bounding box. This makes the metric more informative when evaluating methods for use in real-world surveillance systems. In addition, to support the analysis in Chapter 4, improvements were proposed to Mean Absolute Error to make it independent of the size of objects in the image when evaluating saliency maps.

The next two chapters investigate two further class-agnostic approaches to maritime object detection. The basis for evaluation will be the sequences listed in Table 3.3 in this chapter. After that, in Chapter 6, all the methods are evaluated and compared in the context of a multi-sensor surveillance system for piracy detection. Here, the focus for evaluation will be on the IPATCH sequences from Table 3.3, plus additional IPATCH data.

Chapter 4

Visual Attention and Saliency for Object Detection

4.1 Introduction

The work in this chapter takes inspiration from the power of the human visual system to develop an object detection mechanism for maritime scenes which:

1. can detect objects with no prior expectation of what they are
2. works with different object scales
3. can operate in a range of viewpoints and environmental conditions

This approach is based on the assumption that objects in a maritime surveillance context appear salient compared to their background. Recent Visual Attention, Saliency and Salient Object Detection methods are evaluated to understand which approaches would be the most promising method to form the basis for a maritime object detector. The saliency-based object detection concept is depicted in Fig. 4.1.

The proposed object detection method also incorporates scene context by using horizon detection to infer depth in the scene. This is used to modify the global saliency map so that distant objects can be detected. Finally, temporal filtering is reduce transient salient detections, mirroring the way the human visual system integrates stimuli over time.

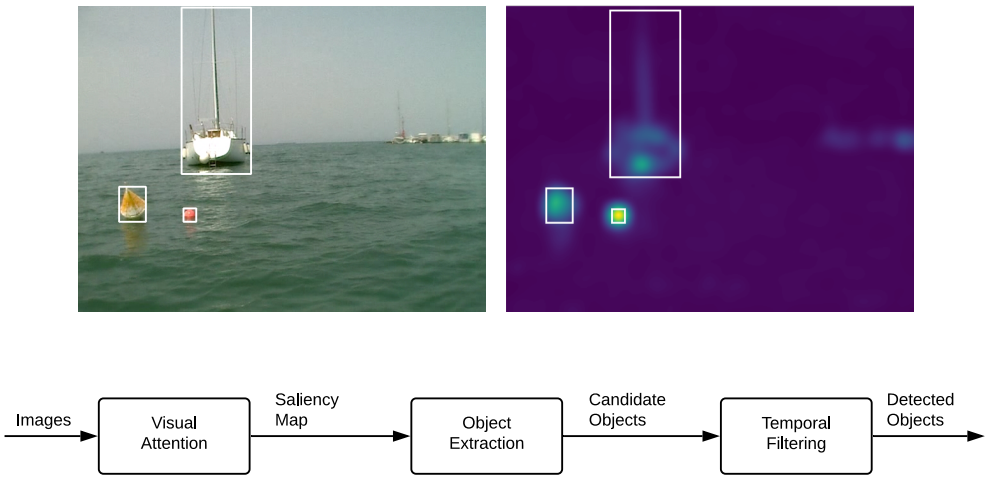


Fig. 4.1 Block diagram showing the saliency-based object detection concept

4.2 Evaluation on maritime surveillance data

Experiments were carried out to analyse the efficacy and utility of the visual attention and saliency methods described in Section 2.3 to see which is the most promising, and to see where improvements could be made. A specific aim was to assess how they perform on the maritime surveillance data that is of interest in this study, and to compare the strengths and weaknesses of the biologically inspired and spectrum analysis approaches.

4.2.1 Experimental set-up

The visual attention and saliency methods from the literature (Table 2.1) were evaluated on a range of maritime surveillance sequences from different publicly available datasets to compare their performance as the basis of an object detection system.

Sequences

A sub-sequence of 500 frames was extracted from selected sequences from a range of datasets. The sequences were chosen to span the range of maritime environments and viewpoints. Some sequences form part of the set chosen for object detection (Table 3.3) but additional sequences were included so as not to bias the selection of saliency method towards specific sequences. Each sub-sequence was chosen so that it included as much variation in object size and appearance as possible, as well as 'empty' frames (frames with



(a) MarDCT wakes-2



(b) MODD 01



(c) SMARTEX Thu-24A-Hitachi



(d) IPATCH 2015-Sc3_Tk2-CAM14



(e) SEAGULL lanchaArgos_clip3



(f) SMD 0797_VIS_OB



(g) SMD 1469_VIS

Fig. 4.2 Example images from the sub-sequences for evaluating saliency performance

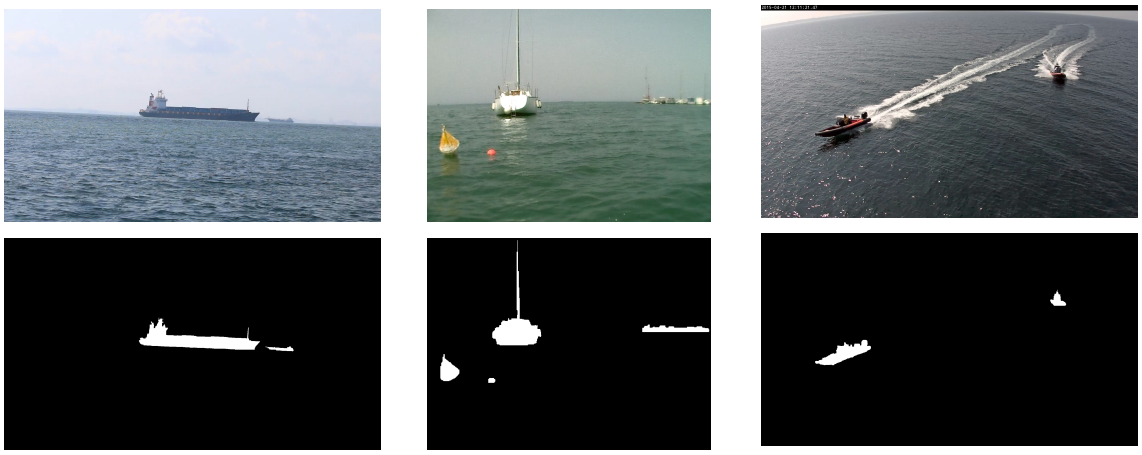
no objects present) in order to assess false positives. The sub-sequences are described in Table 4.1 and some example images are shown in Fig. 4.2.

Groundtruth

Pixel-level groundtruth saliency maps were created for every image in the sub-sequences. Groundtruth saliency maps are binary maps in which salient objects are labelled as 1 and everywhere else is labelled as 0. Initial maps were created by using the bounding box groundtruth for the sequences to initialise a GrabCut process [185]. The saliency maps were then refined to pixel level accuracy by hand.

Table 4.1 Sub-sequences for saliency map evaluation. Key to challenges in Table 3.1

Dataset	Sequence	Sub-sequence frame range	No. of Targets	Target Area pixels (% area)	Challenges
IPATCH	2015-Sc3_Tk2-CAM14	4,901 – 5,400	2	143 – 25,728 (0.007 – 1.241)	EM, MT, DT, SC, W, S/R/W, DS/G
MarDCT	wakes-2	501 – 1,000	1	2 – 573 (0.000 – 0.141)	CM, CZ, LF, ST, DT, W, S/R/W, CA, RA
MODD	01	41 – 540	5	20 – 17,114 (0.007 – 5.571)	EM, MT, SC, S/R/W
SEAGULL	lanchaArgos_clip3	1 – 500	1	33 – 11,605 (0.002 – 0.560)	EM, ST, W, DS/G, V
SMARTEX	Thu-24A-Hitachi	501 – 1,000	2	204 – 3,237 (0.066 – 1.054)	CM, LF, ST, MT, W, S/R/W, RA
SMD	0797_VIS_OB	1 – 500	2	5 – 82,999 (0.000 – 4.003)	CZ, LF, EM, MT, OT, S/R/W
SMD	1469_VIS	101 – 600	11	1 – 67,924 (0.000 – 3.276)	MT, OT, DT, SC, W, S/R/W



(a) SMD 0797_VIS_OB

(b) MODD 01

(c) IPATCH 2015-Sc3_Tk2-CAM14

Fig. 4.3 Examples of groundtruth saliency maps created for the sub-sequences in Table 4.1

Evaluation metrics

For quantitative evaluation of the saliency map output, three metrics were selected from the salient object detection literature to evaluate key areas of performance: Mean Absolute Error (MAE), Precision-Recall (PR) curve, and Receiver Operating Characteristic (ROC) curve [32]. Using both PR and ROC curves is important when dealing with highly skewed datasets [58, 69] such as the PETS IPATCH sequences. The PR and ROC curves for all frames are averaged to create a single curve for the sequence. The area under the curves is also calculated for numerical comparison. In the following definitions, S is the saliency map output by the system and G is the groundtruth saliency map, both in the range $[0,1]$.

Mean Absolute Error (MAE): based on the work in Chapter 3, the proposed balanced version of MAE is used, which is defined as

$$\text{BMAE} = \alpha^{\min(N^-,1)}\text{MAE}^+ + (1-\alpha)^{\min(N^+,1)}\,\text{MAE}^-, \tag{4.1}$$

where

$$\text{MAE}^+ = \frac{1}{N^+}\sum_i\sum_j \left|S^+(i,j) - G^+(i,j)\right| \tag{4.2}$$

and

$$\text{MAE}^- = \frac{1}{N^-}\sum_i\sum_j \left|S^-(i,j) - G^-(i,j)\right| \tag{4.3}$$

To a give a sequence-level score, the BMAE values are averaged over all frames of the sequence:

$$\widehat{\text{BMAE}} = \sum_{n=1}^{N_{frames}} \text{BMAE}(n) \tag{4.4}$$

Use of the proposed BMAE version is important for this work because the objects occupy a very small portion of the image. α is set to 0.5 to weight false positive and false negative errors equally.

Precision-Recall (PR) curves: The Precision-Recall curve plots the fraction of the salient pixels that correspond to salient object regions (Precision) against the fraction of the salient object pixels that were correctly identified in the saliency map (Recall). S is binary thresholded at a range of values, τ, to create a set of binary maps, $\{\bar{S}_\tau\}_{\tau=0}^{1}$. For each $\bar{S}_\tau$, Precision and Recall are calculated as

$$\text{Precision}(\tau) = \frac{|\bar{S}_\tau \cap G|}{|\bar{S}_\tau|} \tag{4.5}$$

$$\text{Recall}(\tau) = \frac{|\bar{S}_\tau \cap G|}{|G|} \tag{4.6}$$

where $|.|$ is the set cardinality operator which denotes the number of pixels in the map equal to 1.

Receiver Operating Characteristic (ROC) curves: The ROC curve, which plots True Positive Rate (TPR) against False Positive Rate (FPR), is calculated in a similar manner using the same set of threshold values.

$$\text{TPR}(\tau) = \frac{|\bar{S}_\tau \cap G|}{|G|} \tag{4.7}$$

$$\text{FPR}(\tau) = \frac{|\bar{S}_\tau \cap G|}{|\bar{S}_\tau \cap G| + |\neg\bar{S}_\tau \cap \neg G|} \tag{4.8}$$

where $\neg$ denotes the inverse of the binary map. Classifiers with points to the left of the ROC curve can be considered more "conservative", whilst those with points to the right can be considered more "liberal" [69].

Implementation

To isolate the role of the saliency method, additional post-processing and data-specific steps (such as using trained classifier features, or learning weights for combining features) were not applied. The exception to this was that background subtraction steps were included for the two papers that use them [189, 212]:

- Tran et al. [212] use a dynamic weighted fusion scheme to combine the saliency map (from MSS saliency [3]) with a foreground mask. The background subtraction step is included in order to compare with Sadhu et al. [189] which uses exactly the same saliency method (MSSS) but with a linear classifier (which could not be recreated).
- Makantasis et al. [144] use an SVM classifier to combine the feature maps and a background subtraction map. However, part of their system is a "Refined Visual Attention Map" which takes the average of the 15 feature maps and multiplies it by the foreground mask from the background subtraction output. This is taken as the saliency map for analysis.

In both works, the background subtraction step dominates the output. This is because they are using static cameras and relatively calm backgrounds. The background subtraction is included in this analysis to see how this approach copes with a wider range of maritime surveillance scenes which involve moving cameras and both static and moving objects to detect.

The original author implementations were used where available and the experiments were run on a 2014 MacBook Pro with 2.6GHz Intel® Core™ i7 processor and 16GB RAM. The DSS method was run on an Alienware laptop with an 8-core 2.6GHz Intel Core i7-6700HQ CPU and 16GB RAM, with an externally connected NVIDIA GeForce GTX Titan X GPU with 12GB memory. Table 4.2 summarises the implementations used.

[1] http://cs-people.bu.edu/jmzhang/BMS/BMS.html
[2] https://github.com/kmakantasis/Poseidon-Features
[3] https://people.cs.umass.edu/~hzjiang/drfi
[4] https://github.com/MingMingCheng/SalBenchmark
[5] https://github.com/Andrew-Qibin/DSS

Table 4.2 Summary of saliency method implementations

Category	Method	ID	Software
Feature Integration	Albrecht [7]	ALB	Own, Python
	Sobral [201]	BMS	BMS Author's [240], C++[1]
	Dawkins [59]	DAW	Own, Python
	Liu [136]	LIU	Own, Python/C
	Makantasis [145]	MAK	Author's, Python[2]
Spectrum Analysis	Ren 2011 [177]	R11	Own, Python
	Ren 2012 [176]	R12	Own, Python
	Ren 2016 [175]	R16	Own, Python
	Sadhu [189]	SAD	Own, Python
	Tran [212]	TRA	Own, Python
	Yao [235]	YAO	Own, Python
Deep Learning	DRFI [222]	DRFI	Author's, C++[3]
	HC [51]	HC	Author's, C++[4]
	DSS [93]	DSS	Author's, Python/Caffe[5]

4.2.2 Results and analysis

The numerical results of the saliency method analysis are presented in Figs. 4.7 - 4.12, and Table 4.3 and Table 4.4. Visual comparison of the saliency maps can be made in Fig. 4.4, 4.5 and 4.6.

Table 4.3 $\widehat{\text{BMAE}}$, AUC PR and AUC ROC results for all methods (best and second best highlighted in green and blue, respectively)

(a) $\widehat{\text{BMAE}}$

Sub-sequence	ALB	BMS	DAW	LIU	MAK	MSS	R11	R12	R16	TRA	YAO	DRFI	HC	DSS
IPATCH_2015-Sc3_Tk2-CAM14	0.4468	0.3777	0.4396	0.3644	0.2242	0.4027	0.4544	0.4298	0.4856	**0.1688**	0.1978	0.3887	0.4539	**0.1439**
MODD_01	0.4127	0.3756	0.4317	0.3854	0.3063	0.4060	0.4511	0.4392	0.4879	**0.2590**	0.3488	0.2896	0.5299	**0.2812**
MarDCT_wakes-2	0.2122	**0.0804**	0.2237	0.2998	0.0926	0.2269	0.1080	0.1775	0.1834	0.1863	**0.0764**	0.1978	0.4092	0.1722
SEAGULL_lanchaArgos-clip3	0.3447	0.2386	0.3412	0.3615	**0.1519**	0.2460	0.4025	0.3224	0.3879	0.1959	0.1933	0.3530	0.2911	**0.0936**
SMARTEX_Thu-24A-Hitachi	0.3253	0.1999	0.3875	0.3912	**0.1479**	0.4388	0.3549	0.4116	0.4538	0.2922	0.2211	**0.1264**	0.4396	0.2743
SMD_MVI_0797_VIS_OB	0.4611	0.4111	0.4844	0.5058	0.3751	**0.3339**	0.4903	0.4946	0.4969	0.3733	0.4302	0.4479	0.3671	**0.2647**
SMD_MVI_1469_VIS	0.4149	0.4338	0.4735	0.3686	0.4714	0.4133	0.4800	0.4673	0.4926	0.4344	0.3840	**0.2643**	0.3251	**0.1943**

(b) AUC PR

Sub-sequence	ALB	BMS	DAW	LIU	MAK	MSS	R11	R12	R16	TRA	YAO	DRFI	HC	DSS
IPATCH_2015-Sc3_Tk2-CAM14	0.0203	0.1688	0.0481	0.1260	**0.2733**	0.0622	0.0040	0.0458	0.0318	0.0182	**0.2556**	0.1417	0.0034	0.1742
MODD_01	0.4204	0.4770	0.4441	0.3852	0.4836	0.6260	0.1751	0.4613	0.1781	0.5558	**0.6698**	0.5274	0.0349	**0.6648**
MarDCT_wakes-2	0.0289	**0.3224**	0.1091	0.0104	0.1846	0.0138	0.2878	0.1043	0.1599	0.0517	0.2869	0.1277	0.0006	**0.4170**
SEAGULL_lanchaArgos-clip3	0.0764	**0.4892**	0.1012	0.0686	0.1820	0.3276	0.0062	0.0888	0.0849	0.1462	0.2619	0.2639	0.0090	**0.6963**
SMARTEX_Thu-24A-Hitachi	0.1997	0.3315	0.2286	0.1120	**0.5272**	0.3192	0.1537	0.2186	0.1978	0.1247	0.4945	0.4493	0.0068	**0.7294**
SMD_MVI_0797_VIS_OB	0.0775	0.5643	0.0600	0.0524	0.2471	**0.7104**	0.0842	0.0348	0.0397	0.1593	0.2663	0.3758	0.0899	**0.7752**
SMD_MVI_1469_VIS	0.6892	0.8280	0.2898	0.5381	0.1527	0.8059	0.2795	0.5579	0.2825	0.3771	0.4524	**0.8486**	0.3034	**0.8324**

(c) AUC ROC

Sub-sequence	ALB	BMS	DAW	LIU	MAK	MSS	R11	R12	R16	TRA	YAO	DRFI	HC	DSS
IPATCH_2015-Sc3_Tk2-CAM14	0.8782	0.9845	0.9519	0.9333	0.9411	0.9674	0.5777	0.9318	0.6977	0.8088	**0.9927**	0.9289	0.6889	**0.9887**
MODD_01	0.8606	**0.9239**	0.8855	0.8159	0.7796	**0.9371**	0.8088	0.8318	0.6988	0.7285	0.9228	0.8512	0.5625	0.8577
MarDCT_wakes-2	0.9829	**0.9993**	0.9919	0.9733	0.8703	0.9313	0.9988	0.9973	0.9924	0.6344	**0.9992**	0.9913	0.6890	0.8893
SEAGULL_lanchaArgos-clip3	0.9711	**0.9978**	0.9829	0.9631	0.9896	0.9932	0.6165	0.9808	0.8923	0.8339	**0.9938**	0.9519	0.8578	0.9889
SMARTEX_Thu-24A-Hitachi	0.9637	**0.9958**	0.9322	0.9600	0.8641	0.9292	0.8346	**0.9919**	0.9278	0.6538	0.9880	0.9900	0.8039	0.9414
SMD_MVI_0797_VIS_OB	0.7911	**0.9092**	0.7656	0.6529	0.8052	**0.9501**	0.6569	0.6755	0.6300	0.6520	0.7275	0.6663	0.7878	0.8880
SMD_MVI_1469_VIS	0.9471	**0.9703**	0.8229	0.9194	0.5371	**0.9650**	0.8005	0.9071	0.7575	0.5663	0.9429	0.9433	0.7533	0.9291

Results for biologically-inspired methods (Figs. 4.7 and 4.11): MAK performs the best over all sequences, followed closely by BMS, when considering the BMAE score (lower is better). However, on the PR and ROC curves, BMS gets slightly better scores than MAK. The background subtraction step used in MAK has a noticeable effect in some sequences. In the SMARTEX_Thurs_24A_Hitachi sequence, there is a significant decrease in performance (increase in BMAE) corresponding to a large camera movement around frame 900. In contrast, BMS (and other methods) are robust to this. Similarly, in the SMD_1469 sequence, MAK achieves a poor ROC curve due to it learning the static ships into the background model. No method performs particularly well on the SMD_1469

sequence; there are many, larger objects competing for saliency so their regions are not fully localised, meaning that the MAE^+ error is high.

Results for frequency analysis methods (Figs. 4.8 and 4.12): YAO performs best overall in BMAE, PR and ROC, with TRA also performing well. However, TRA is affected by camera motion (e.g. SEAGULL and SMD_0797 sequences) due to the background subtraction step. The background subtraction step also means that TRA has similar PR and ROC characteristics to MAK, in that the curves contain jumps caused by the large number of zeros present in the saliency map. Whilst MSS does not achieve good BMAE scores, it suddenly becomes competitive under PR and ROC curve analysis. Similar to the biologically-inspired methods, all the frequency analysis methods struggle with the SMD_1469 sequence. Here, the low performance is due to the assumption of high frequency noise in the image and the mismatch with object size.

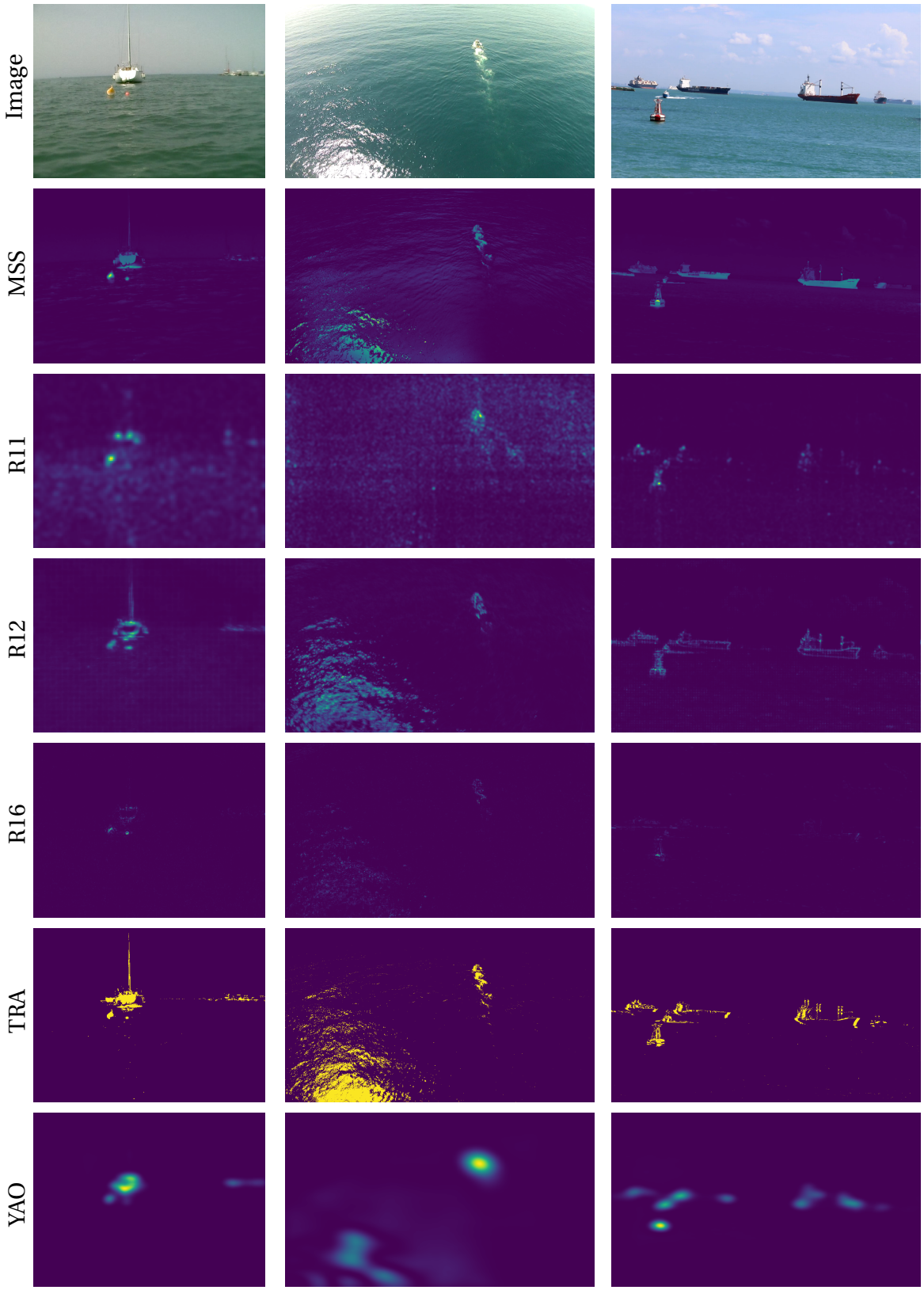
Results for salient object detection methods (Figs. 4.9 and 4.13): Looking at the BMAE scores, it can be seen that DRFI over-estimates (predicts false positives) empty, or nearly empty, scenes (e.g. start of IPATCH, parts of SEAGULL). This is because it has been trained on images where there is always at least one salient object present. DRFI and DSS make errors when the object is near the edge of the frame. Again, this is due to the training data used, in which objects do not often appear at the edges of the images (the so-called 'centre bias'). Whilst HC is a faster method, it does not perform as well as DSS and DRFI in terms of PR or ROC curves. It is not robust to colour / contrast changes in the image (e.g. the 'wave' pattern in SMD_0797 is due to image brightness (colour) changes caused by camera motion). DSS is the best method overall.

Comparison of the top performing methods (Figs. 4.10 and 4.14): Under BMAE, DSS is best overall, but it should be noted that it is not universally better (i.e. for some sequences and under some conditions, other methods have lower error). DSS gets better PR curves overall, with BMS a close second. However, BMS gets better ROC curves. The area under the curve (AUC) of an ROC curve (see Table 4.4 (c)) is equivalent to the probability that the classifier will rank a randomly chosen positive instance higher than a randomly chosen negative instance [69]. The discrimination power between salient and non-salient regions is therefore higher under BMS.



(a) MODD 01 (b) SEAGULL lanchaArgos (c) SMD MVI_1469_VIS

Fig. 4.4 Example saliency maps from biologically-inspired methods



(a) MODD 01 (b) SEAGULL lanchaArgos (c) SMD MVI_1469_VIS

Fig. 4.5 Example saliency maps from frequency analysis methods



Fig. 4.6 Example saliency maps from salient object detection methods

Other observations: The sharp transition in some of the ROC curves (e.g. for the SMARTEX sequence in Fig. 4.14) are caused when a method outputs a large number of 0 values in its saliency map. Because of the thresholding process used to create these curves, a value of 0 goes from a false positive to either a true negative or false negative at the lowest threshold point. This creates a large drop in the number of false positives, which creates a jump in the false positive rate / recall value. This is especially the case in the MAK method (due to the background subtraction masking step which sets many values to 0) and the DSS method (which correctly predicts a lot of low saliency values). Methods with a smoother distribution have smoother PR and ROC curves.

Conclusion: Due to its high performance in recent salient object detection [33] and gaze fixation benchmarks [42, 34], its real-time speed, and its performance in the above evaluation on maritime surveillance data, BMS is selected as the most promising basis on which to build a maritime object detector. BMS is also attractive as it does not involve decisions about how to combine the maps (e.g. selecting or learning weights) — it is a complete solution in that sense. In the following sections, modifications to the baseline BMS approach are explored to tailor it for the task of maritime object detection.

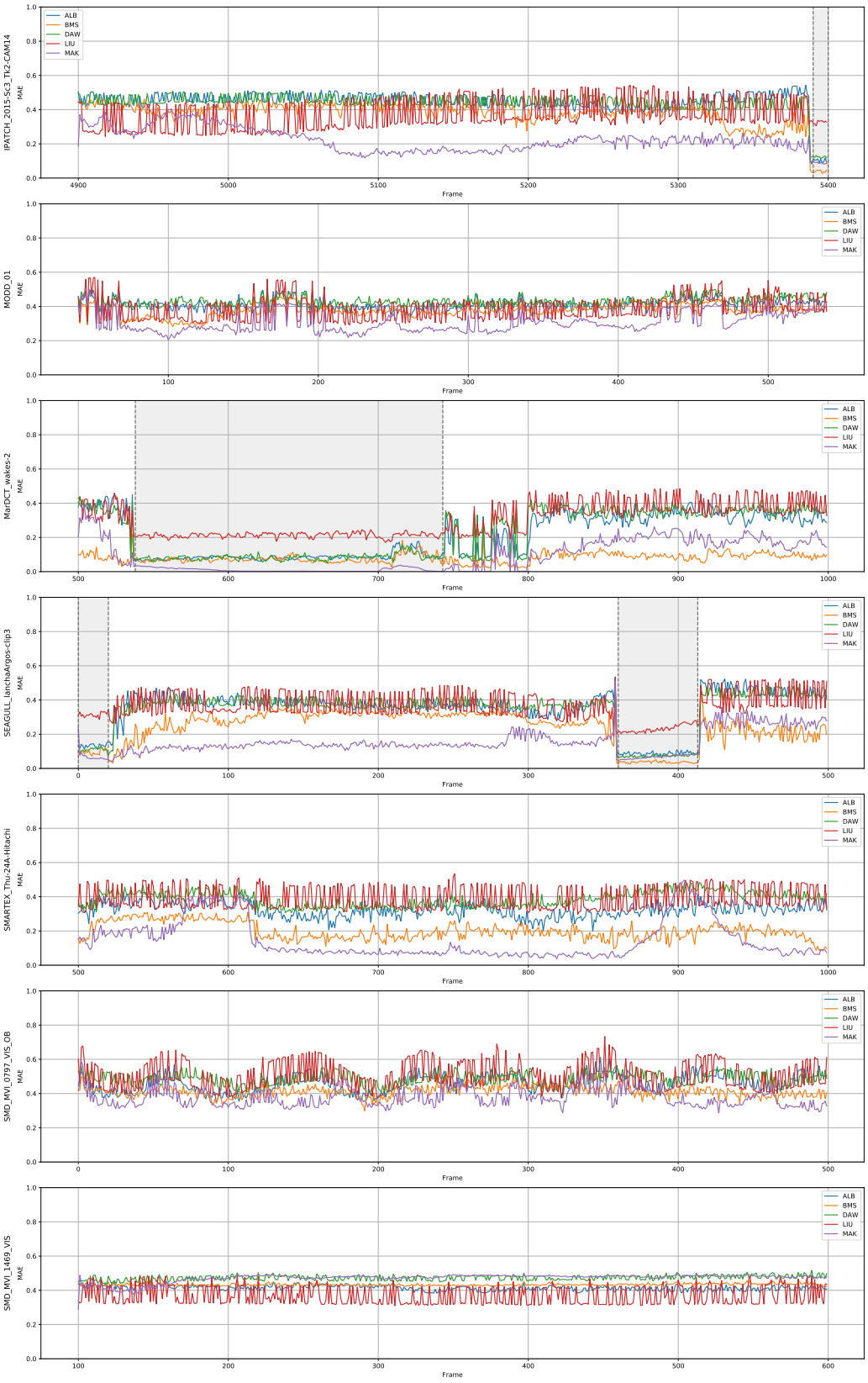


Fig. 4.7 $\widehat{\text{BMAE}}$ vs. frame number for biologically-inspired approaches

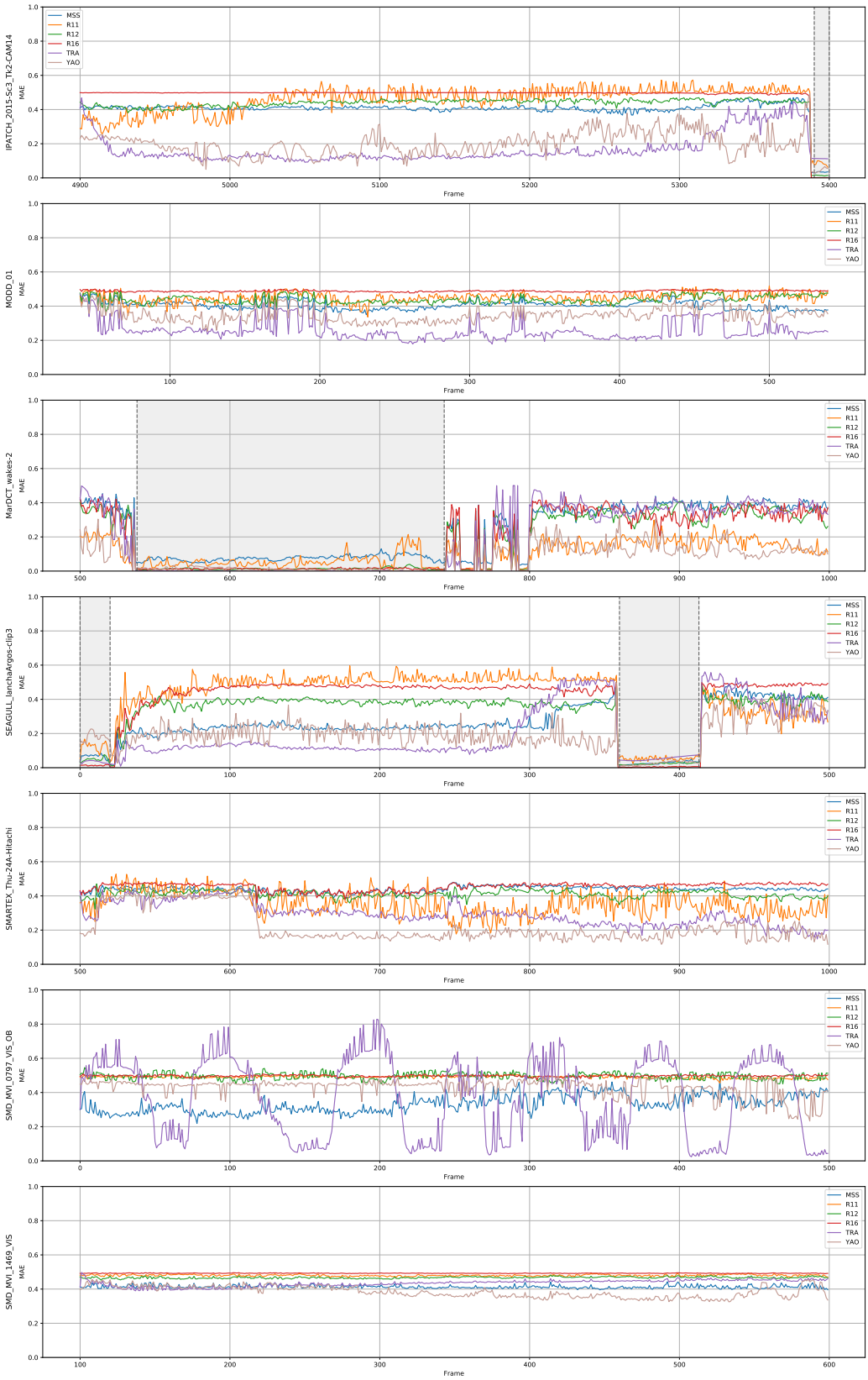


Fig. 4.8 $\widehat{\text{BMAE}}$ vs. frame number for spectrum analysis approaches



Fig. 4.9 $\widehat{\text{BMAE}}$ vs. frame number for salient object detection approaches

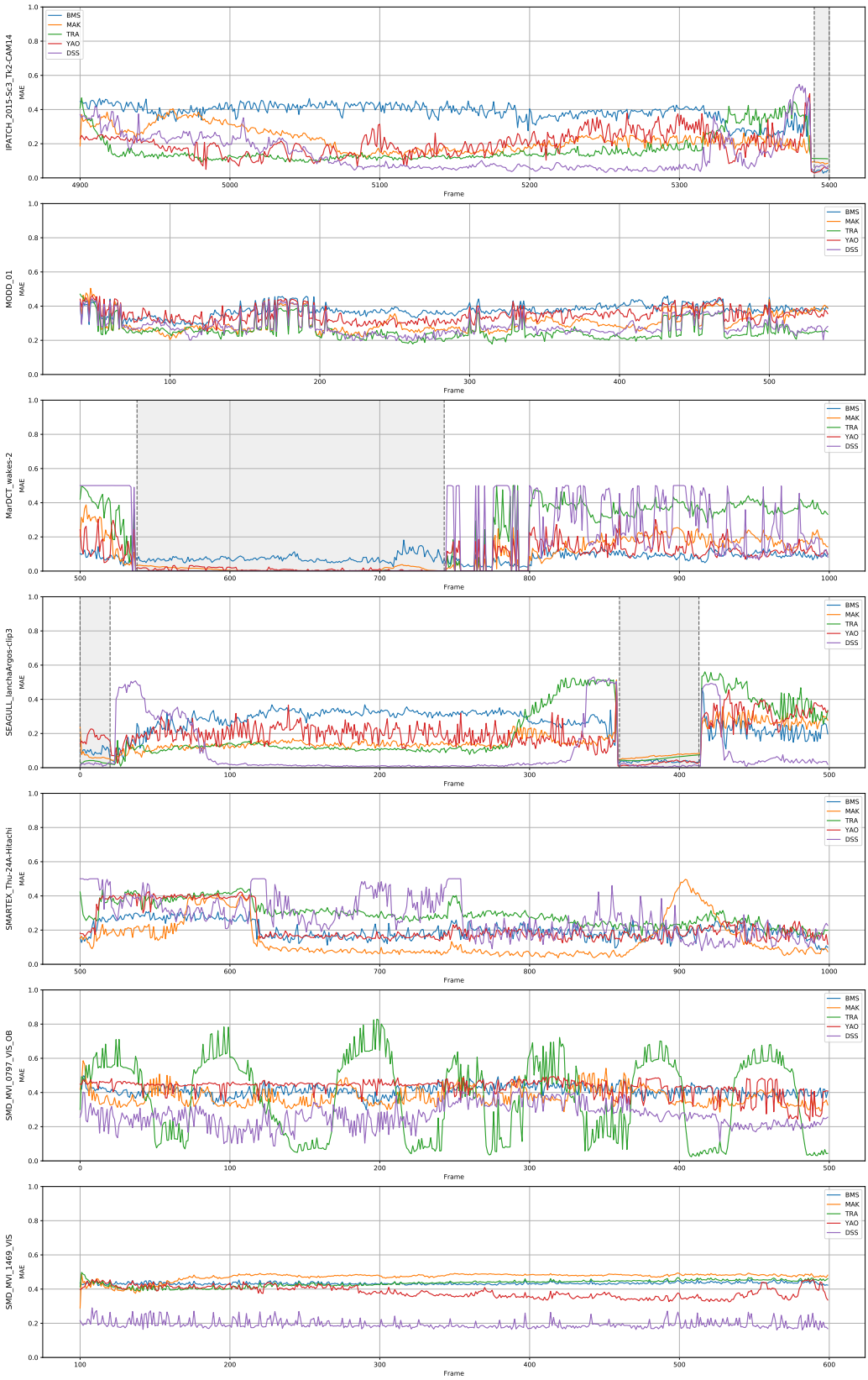


Fig. 4.10 $\widehat{\text{BMAE}}$ vs. frame number for the best approaches from each category

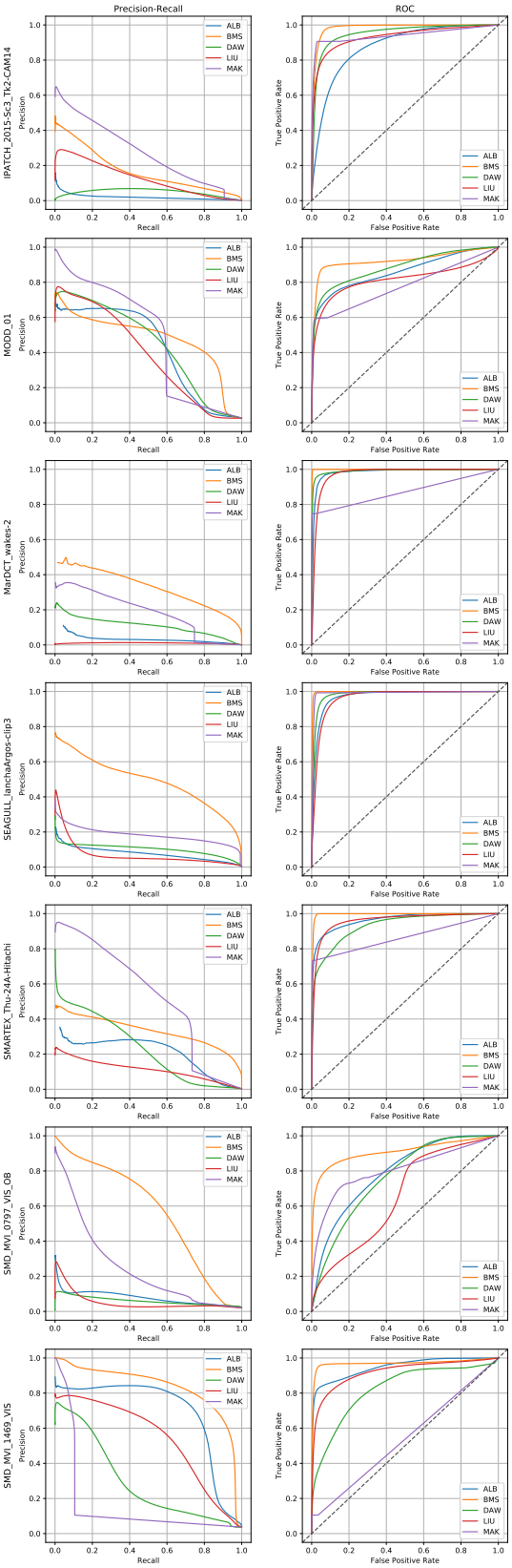


Fig. 4.11 PR and ROC curves for biologically-inspired approaches

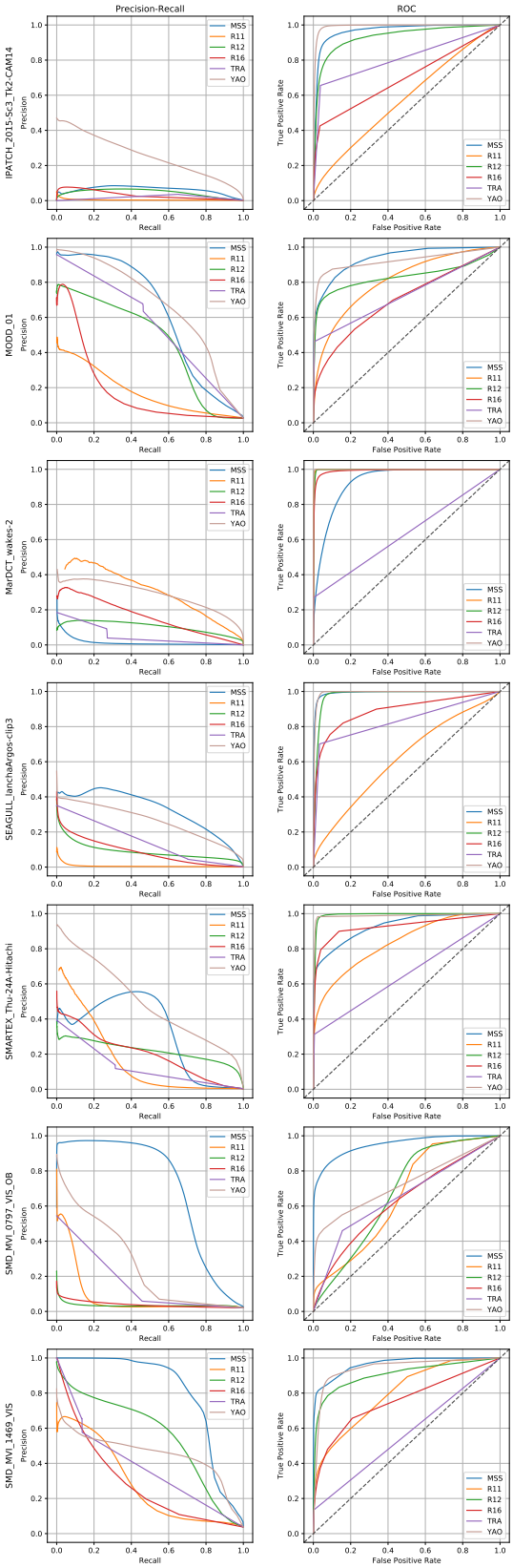


Fig. 4.12 PR and ROC curves for spectrum analysis approaches

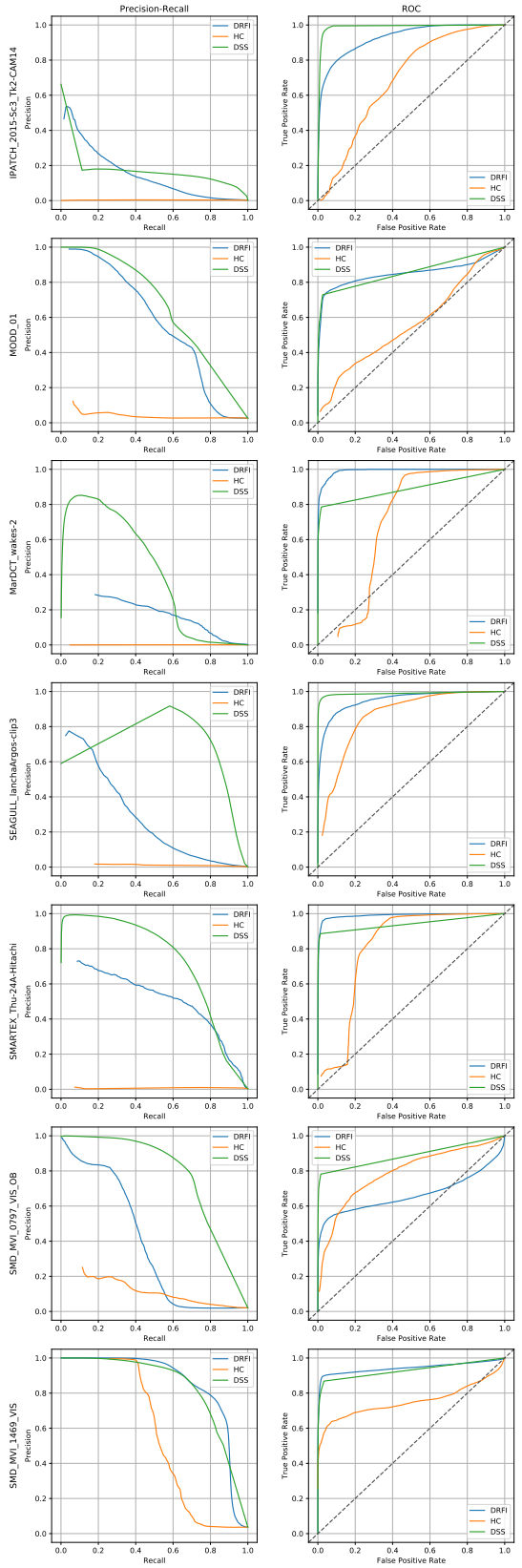


Fig. 4.13 PR and ROC curves for salient object detection approaches

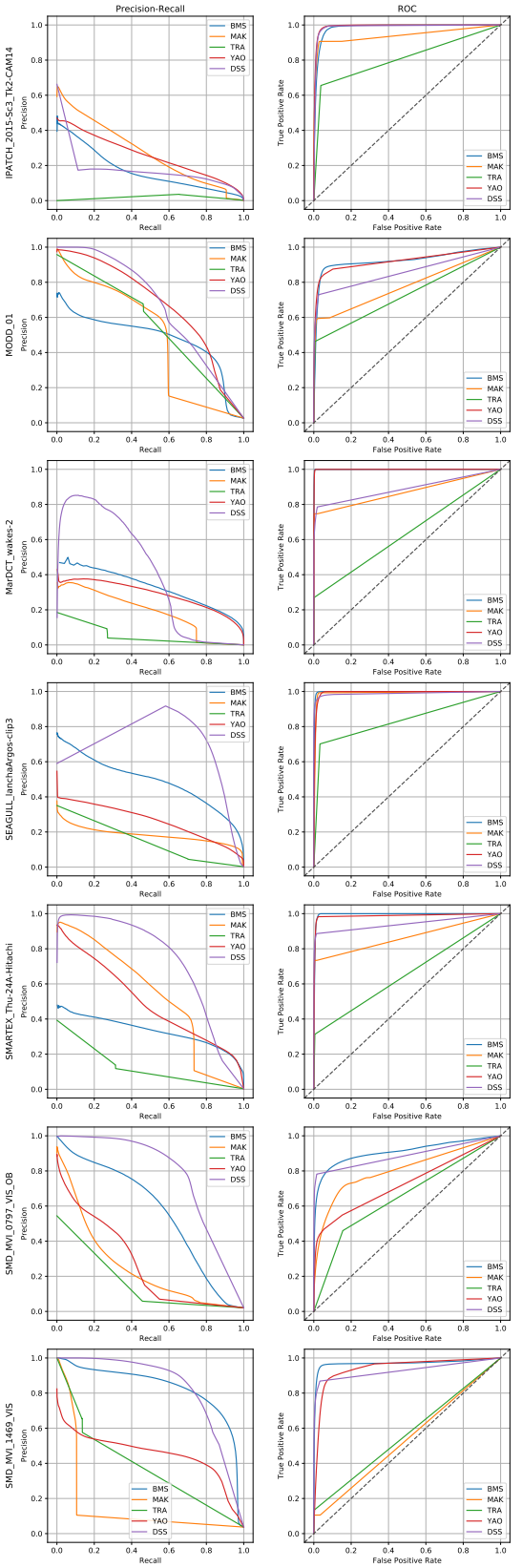


Fig. 4.14 PR and ROC curves for the best approaches from each category

Table 4.4 $\widehat{\text{BMAE}}$, AUC PR and AUC ROC results for the top performing methods from each category (best and second best highlighted in green and blue, respectively)

(a) $\widehat{\text{BMAE}}$

Sub-sequence	BMS	MAK	TRA	YAO	DSS
IPATCH_2015-Sc3_Tk2-CAM14	0.3777	0.2242	**0.1688**	0.1978	**0.1439**
MODD_01	0.3756	0.3063	**0.2590**	0.3488	**0.2812**
MarDCT_wakes-2	**0.0804**	0.0926	0.1863	**0.0764**	0.1722
SEAGULL_lanchaArgos-clip3	0.2386	**0.1519**	0.1959	0.1933	**0.0936**
SMARTEX_Thu-24A-Hitachi	**0.1999**	**0.1479**	0.2922	0.2211	0.2743
SMD_MVI_0797_VIS_OB	0.4111	0.3751	**0.3733**	0.4302	**0.2647**
SMD_MVI_1469_VIS	0.4338	0.4714	0.4344	**0.3840**	**0.1943**

(b) AUC PR

Sub-sequence	BMS	MAK	TRA	YAO	DSS
IPATCH_2015-Sc3_Tk2-CAM14	0.1688	**0.2733**	0.0182	**0.2556**	0.1742
MODD_01	0.4770	0.4836	0.5558	**0.6698**	**0.6648**
MarDCT_wakes-2	**0.3224**	0.1846	0.0517	0.2869	**0.4170**
SEAGULL_lanchaArgos-clip3	**0.4892**	0.1820	0.1462	0.2619	**0.6963**
SMARTEX_Thu-24A-Hitachi	0.3315	**0.5272**	0.1247	0.4945	**0.7294**
SMD_MVI_0797_VIS_OB	**0.5643**	0.2471	0.1593	0.2663	**0.7752**
SMD_MVI_1469_VIS	**0.8280**	0.1527	0.3771	0.4524	**0.8324**

(c) AUC ROC

Sub-sequence	BMS	MAK	TRA	YAO	DSS
IPATCH_2015-Sc3_Tk2-CAM14	0.9845	0.9411	0.8088	**0.9927**	**0.9887**
MODD_01	**0.9239**	0.7796	0.7285	**0.9228**	0.8577
MarDCT_wakes-2	**0.9993**	0.8703	0.6344	**0.9992**	0.8893
SEAGULL_lanchaArgos-clip3	**0.9978**	0.9896	0.8339	**0.9938**	0.9889
SMARTEX_Thu-24A-Hitachi	**0.9958**	0.8641	0.6538	**0.9880**	0.9414
SMD_MVI_0797_VIS_OB	**0.9092**	0.8052	0.6520	0.7275	**0.8880**
SMD_MVI_1469_VIS	**0.9703**	0.5371	0.5663	**0.9429**	0.9291

4.3 Creating an object detector for maritime surveillance

The proposed object detection system (Fig. 4.15) creates a saliency map for each frame using the BMS method. Thresholding is performed on the saliency map to locate the salient regions corresponding to potential objects. The list of candidate objects is filtered over a short period of time to remove transient detections and smooth the estimates of object location and size.



Fig. 4.15 Block diagram presenting the different stages of the proposed algorithm.

4.3.1 Boolean Map Saliency

The Boolean Map Saliency (BMS) method [241] (Fig. 4.16) exploits the visual property of 'surroundedness' whereby objects in an image are more salient, the more surrounded they are by background regions in a given feature space. In principle, any feature channels can be used (colour, orientation, motion, etc.), but the method in [241] found the CIELAB colour channels to be the best for natural images.

First, the image is transformed into the CIELAB colourspace and the colourspace is rectified using a whitening step:

$$\bar{\mathbf{x}} = \frac{1}{n}\sum_i \mathbf{x}_i \tag{4.9}$$

$$\mathbf{Q} = \frac{1}{n}\sum_i \mathbf{x}_i \mathbf{x}_i^T - \bar{\mathbf{x}}\bar{\mathbf{x}}^T \tag{4.10}$$

$$\mathbf{y}_i = (\mathbf{Q} + \lambda \mathbf{I})^{-\frac{1}{2}} \cdot \mathbf{x}_i \tag{4.11}$$

This maps each of the L, A, B channels to their normalised and decorrelated counterparts, L′, A′ and B′. Each of the channels, L′, A′ and B′, are then normalised to the range $[0,255]$ and median filtered[6]. The channels are then binary thresholded at intervals with a step size of δ, yielding a set of N binary images (Boolean maps), $\{B_i\}_{i=1}^{N}$.

[6]The median filtering step is not mentioned in either paper [240, 241] but was found in the author's published code.

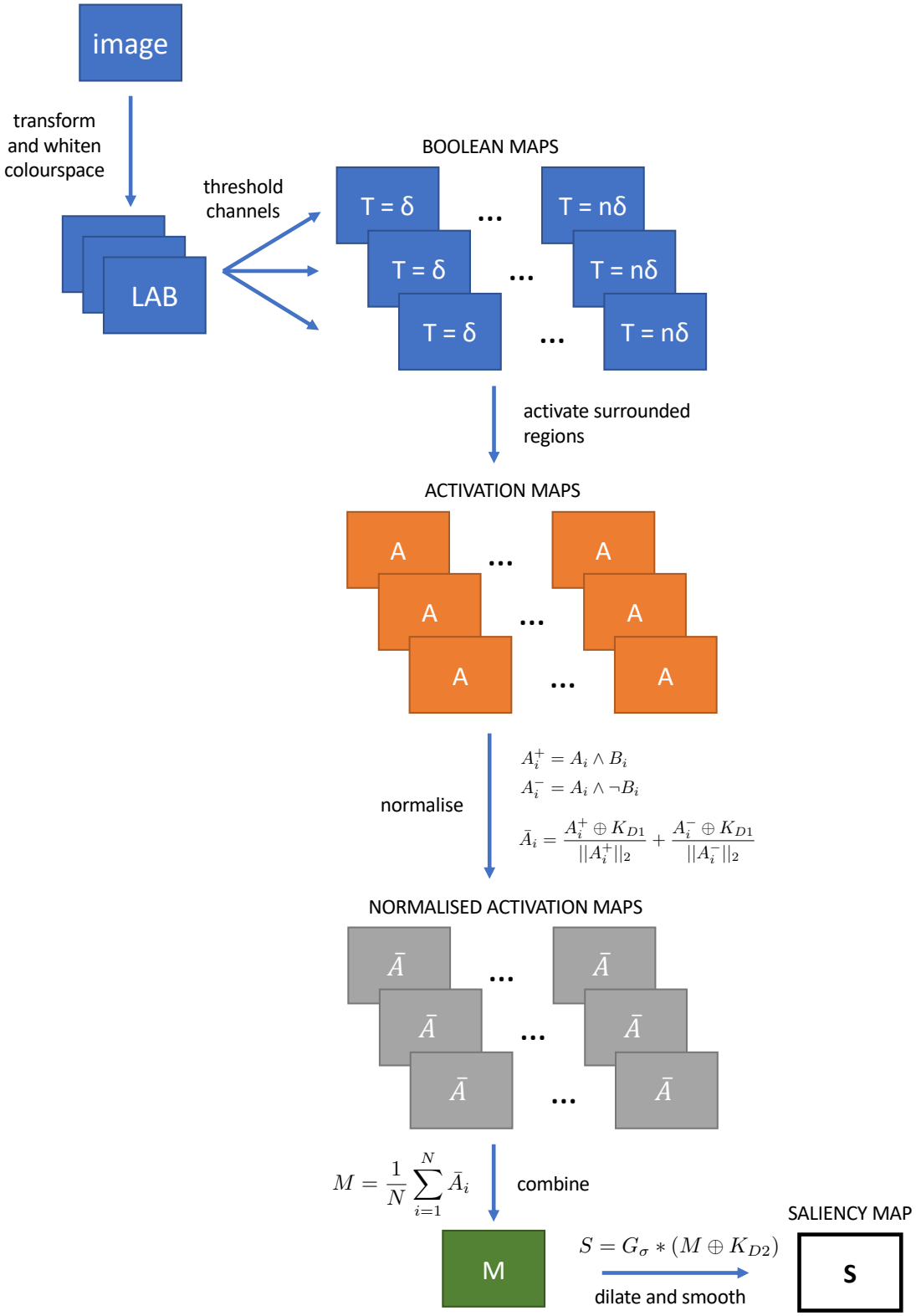


Fig. 4.16 Stages of the BMS pipeline

An activation map is then created for each Boolean map by identifying the surrounded regions. A black region is surrounded in B_i if it is enclosed by a white region and *vice versa*. The activation map, A_i, is created by setting pixels to 1 if the corresponding pixel is in a surrounded region of B_i, and setting 0 elsewhere. The set of activation maps, $\{A_i\}_{i=1}^{N}$, is then normalised in order to emphasise maps with small activated regions. First, each activation map is split into two sub-activation maps, A_i^+ and A_i^-, according to

$$A_i^+ = A_i \wedge B_i, \tag{4.12}$$

$$A_i^- = A_i \wedge \neg B_i, \tag{4.13}$$

where $\wedge$ represents pixel-wise logical AND between two binary maps and $\neg B_i$ is the negation (logical NOT) of B_i. Both sub-activation maps are dilated with a square kernel K_{D1} of size $D1$ and divided by their L2-norm. This serves to emphasise clumps of small activated regions whilst reducing the importance of small, scattered regions. The normalised activation map, $\bar{A}_i$, is therefore calculated as

$$\bar{A}_i = \frac{A_i^+ \oplus K_{D1}}{||A_i^+||_2} + \frac{A_i^- \oplus K_{D1}}{||A_i^-||_2}, \tag{4.14}$$

where $\oplus$ represents the morphological dilation operation. The final saliency map, S, is found by taking the average of all the normalised activation maps and performing a second dilation operation followed by Gaussian smoothing:

$$M = \frac{1}{N} \sum_{i=1}^{N} \bar{A}_i, \tag{4.15}$$

$$S = G_\sigma * (M \oplus K_{D2}), \tag{4.16}$$

where K_{D2} is a square dilation kernel of size $D2$ and G_σ is a Gaussian kernel with standard deviation σ. The results of the different stages of the BMS pipeline are shown in Fig. 4.17.

The algorithm parameters are set to the values used by the author [241] ($\delta = 8$, $D1 = 7$, $D2 = 9$ and $\sigma = 9$). These have been determined empirically and tests showed that other values did not significantly improve performance in the maritime case. The exception is the value of δ, which has an impact on speed. This is examined further in the experiments in this chapter.



(a) B_i (b) $\neg B_i$ (c) A_i^+ (d) A_i^- (e) $\bar{A}_i$



(f) M (g) S

Fig. 4.17 Example maps from each stage of the BMS pipeline

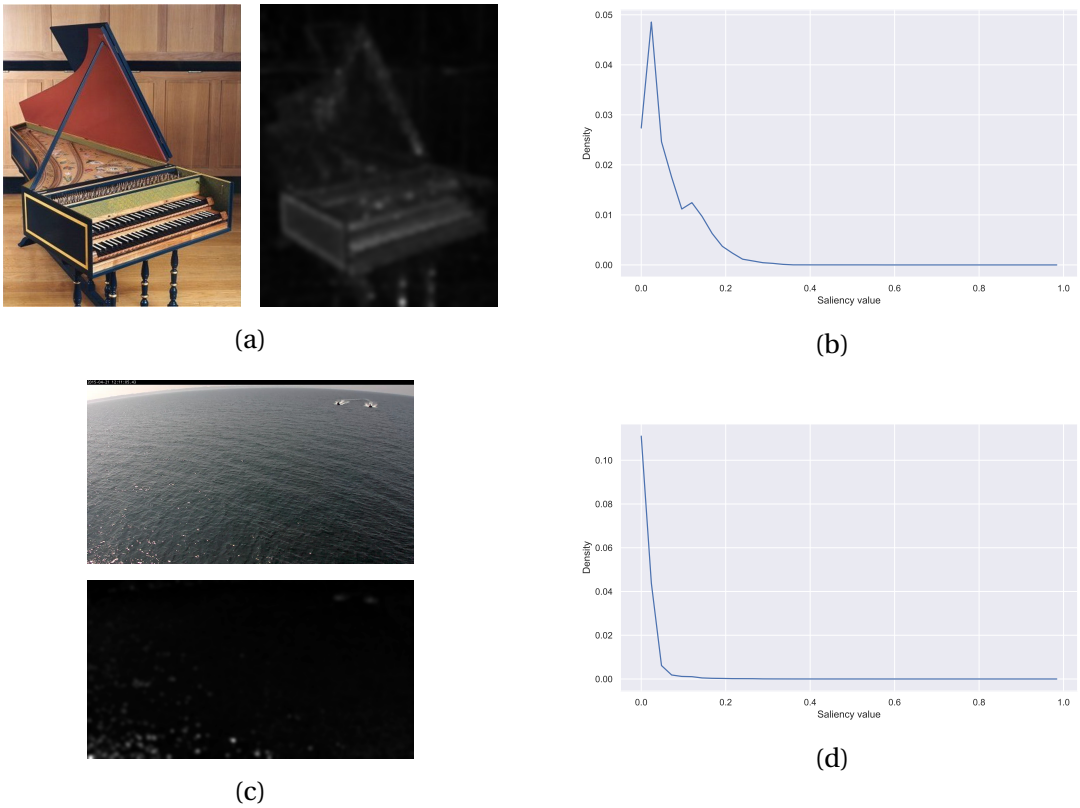
4.3.2 Candidate region extraction

The most common way to extract object regions from a saliency map is binary thresholding followed by connected components analysis. Obviously, a single, fixed threshold value would not generalise well across different images or datasets, so most approaches adopt an adaptive method, in which the threshold value changes according to global or local image properties (i.e. driven by the data itself).

The threshold level is often tied to a multiple of the mean saliency value [2, 94, 108], most commonly two times the mean [125]. There is no principled reason behind these choices; most likely they are values which give good results on the benchmark datasets. Setting the threshold as an arbitrary multiple of the mean value really just displaces the problem of selecting a fixed threshold to a problem of how to choose the best multiple instead.

The choice of a multiple of the mean is inherently linked to the shape of the distribution of the saliency map. Fig. 4.18 shows example distributions from a salient object detection benchmark dataset image and a maritime surveillance image. In the salient object benchmarks, objects occupy a reasonable proportion of the image (see Fig. 4.19), so the assumption that the mean will be a good discriminator is valid. However, looking at the distribution of the saliency map from the maritime image, it can be seen that the distribution is more heavily weighted to the lower end of values. This is because maritime objects are sparse and much smaller compared to the background class.

Methods such as Otsu [157] and triangle [238] thresholding use the distribution of saliency values to split into two classes. These methods were tried on the maritime sequences in this work, but their performance was not satisfactory. The implicit assumption that the image contains saliency values drawn from two modes of a distribution of roughly the same size does not hold.



(a) (b) (c) (d)

Fig. 4.18 Example images, BMS saliency maps and saliency value distributions for a salient object detection benchmark image from [107] and a maritime surveillance image from IPATCH.

An ideal saliency map output for a scene with few small salient objects would have a very large number of 0s and a very small number of 1s. As an example, an object of 10,000 pixels in a 1920 × 1080 image would occupy < 1%. This means that the mean value is very close to zero (in practice, it is often zero because of numerical precision). The proposed approach therefore uses thresholding based on the top *percentiles* of saliency values, rather than the mean.

The threshold is set to a high percentile (e.g. 99th) of the saliency map. This captures the most salient points in the image, but is likely to miss true object regions which were



Fig. 4.19 Distributions of object size (represented as proportion of total image) for the MSRA-B dataset [107] and the maritime surveillance sub-sequences (Table 4.1). In salient object detection datasets, there is a wider spread of object sizes covering a reasonable proportion of the image. In maritime surveillance datasets, objects tend to occupy a very small proportion of the image.

still highly salient but not in the top 1%. However, a lower threshold is likely to introduce more false detections.

Hysteresis thresholding is a common way to address this and is used here for this purpose, as it has been in other recent maritime works [117, 142]. Two thresholds are set; an upper and a lower. The saliency map is binary thresholded at the upper value and the flood-fill algorithm is then used to grow regions to add connected pixels which are above the lower threshold.

Although the upper and lower percentile values must be fixed by the user, the approach can still be regarded as adaptive, as the distribution of the data determines the threshold value in each image. The proposed thresholding method is named *adaptive hysteresis thresholding*.

Note that using a percentile-based threshold assumes that there will be a minimum number of salient pixels in the image. Clearly, it would be overly optimistic to assume that some salient object will always be present. The assumption being made here is that, if salient pixels in the image are coherently grouped over a number of frames, then they are likely to be an object. In scenes where there are no objects of interest, the assumption is that the salient pixels will be dispersed in small, transient regions caused by reflections and other sea motion. These can be filtered out in a secondary step, which is explained in Section 4.3.4.

4.3.3 Mitigating saliency of wake through horizontal and vertical thresholding

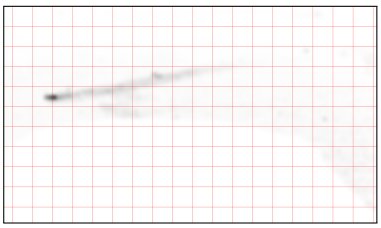
With adaptive hysteresis thresholding applied to the whole image, the system can detect boat objects but also the connected wake regions, leading to large bounding boxes which fully contain the true target but overestimate its size. Fundamentally, wake is very salient in maritime images as it presents a bright, white region surrounded by a contrasting background of the sea. However, boats are often white or light in colour, so suppressing white regions in the image as done in [28] is not a good approach in all situations.

A modification to the thresholding step is proposed to reduce detection of wake by exploiting the fact that the boat is locally salient, relative to the wake (see Fig. 4.20). By looking at limited regions of the image one at a time – in this case horizontal and vertical strips – different parts of the image will appear in the percentile range for the hysteresis thresholding step than if the whole image is taken into account. Large wake regions dominate the upper part of the saliency distribution globally, but considering smaller regions forces the wake to compete with itself. This has the effect of fragmenting the wake. By only extracting regions which are salient in more than one axis, a lot of the fragments are filtered out. In this study, a width of 100 pixels is used to create the strips.

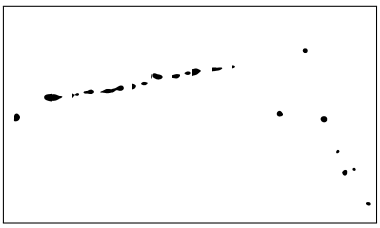
4.3.4 Reducing false positives through temporal filtering

Temporal averaging of the saliency map was tested, but it did not have a significant impact on performance. As with background subtraction methods, it is difficult to choose the size of the time window appropriately for all sequences. A bigger problem was that it enhances non-object regions and transient/noisy detections in the saliency map, rather than suppressing them. Saliency maps do not exhibit the same properties as intensity images, so transient salient points can dominate the average and accumulate over time, swamping the true target regions. Temporal filtering of detections was found to work better. This approach favours regions which are persistently above the detection threshold (rather than a pixel-level average of saliency values, which does not reflect global saliency).

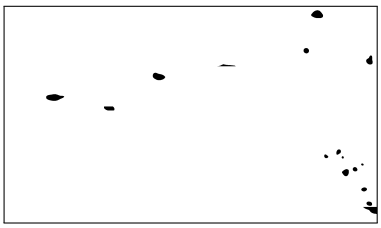
A simple frame-to-frame tracking framework was implemented to filter out transient detections. In each frame, new detections are assigned to tracked detections from the previous frame using the Hungarian algorithm [124, 150]. The cost matrix is completed by calculating the overlap error (= 1 − IoU) between each pair of bounding boxes. Gating



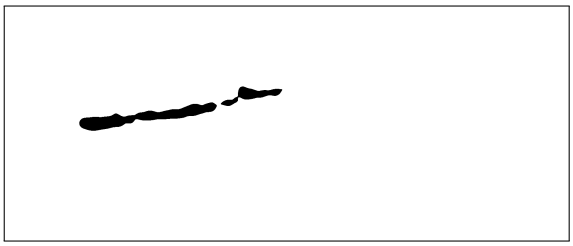
(a) Saliency map analysed in horizontal and vertical strips



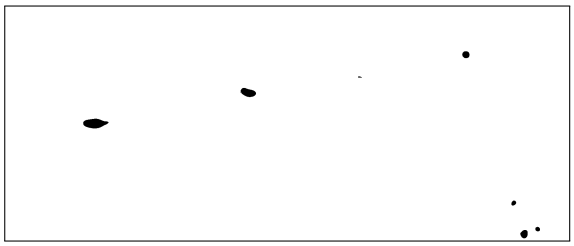
(b) Result of thresholding in horizontal strips



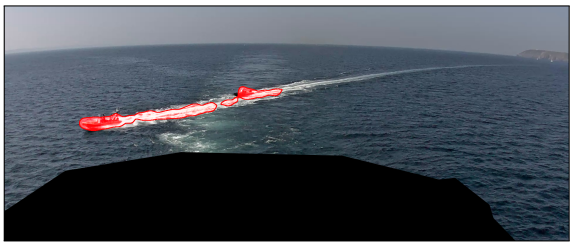
(c) Result of thresholding in vertical strips



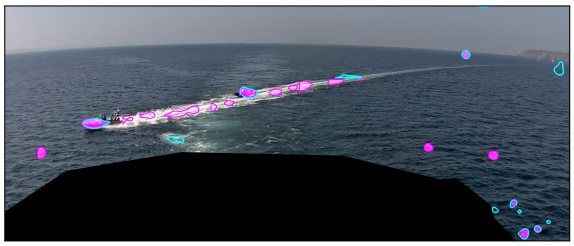
(d) Global hysteresis thresholding



(e) Horizontal-vertical hysteresis thresholding



(f) Salient regions using global hysteresis thresholding



(g) Salient regions using horizontal-vertical hysteresis thresholding

Fig. 4.20 Visualisation of horizontal-vertical thresholding. With a global thresholding approach (d, f), the large wake region dominates the upper part of the distribution of saliency values. By considering limited regions of the image through horizontal and vertical strips (e, g), the wake can be suppressed. In (f), the regions of the image extracted by thresholding are overlaid in red. In (g), the regions of the image extracted by horizontal and vertical thresholding are overlaid in magenta and cyan, respectively.

is implemented by introducing a maximum cost threshold for assignment, d_{max}, such that matches are discarded if the overlap error is greater than d_{max}. Matches between detections in two consecutive frames triggers the creation of a new track which is managed by a standard nearly-constant-velocity Kalman filter [111] with the following state space and process models:

$$\mathbf{x} = \begin{bmatrix} x_c & y_c & \dot{x}_c & \dot{y}_c & w & h & \dot{w} & \dot{h} \end{bmatrix}^T \tag{4.17}$$

$$\mathbf{x}_k = \mathbf{F}\mathbf{x}_{k-1} + \mathbf{v}_k \tag{4.18}$$

$$\mathbf{z}_k = \mathbf{H}\mathbf{x}_k + \mathbf{w}_k \tag{4.19}$$

$$\mathbf{v}_k \sim \mathcal{N}(0, \mathbf{Q}) \tag{4.20}$$

$$\mathbf{w}_k \sim \mathcal{N}(0, \mathbf{R}) \tag{4.21}$$

where (x_c, y_c) and $(\dot{x}_c, \dot{y}_c)$ are the position and velocity of the bounding box centroid, and w, h, $\dot{w}$ and $\dot{h}$ are the width and height of the bounding box and their respective rates of change. The transition and observation matrices, **F** and **H**, are taken as

$$\mathbf{F} = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \qquad \mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \tag{4.22}$$

Observation noise is assumed to be 1 pixel, which may seem low for a visual detection method. However, the quantity being measured is the underlying saliency map, rather than the object in the image domain. When it is constructed, the saliency map undergoes discretisation (boolean maps) and filtering (dilation and Gaussian smoothing), so it is less susceptible to noise than pixels in the image domain. It is assumed that the height and width of the salient region are uncorrelated with position in the image.

The process is modelled as small, nearly-constant-velocity motions from frame to frame (i.e. acceleration is not zero, but it is noisy). Process noise covariance is therefore assumed to be small. The observation noise covariance, **R**, was initialised with the identity matrix and the process noise covariance, **Q**, was tuned empirically. These values can undoubtedly be optimised further under different conditions. However, the focus of this work is detection and the filtering step is targeted at very short-term tracking to filter out false positives. The resulting detections will then be passed to a more sophisticated tracking stage in Chapter 6.

When new detections are assigned to existing tracks, the track is updated by estimating the state using the new observation. Tracks above a minimum length L are output as stable detections. If a track is not assigned a new detection in the frame, the new bounding box is predicted by the Kalman filter. The filter is allowed to predict up to T frames without a new matched detection before the track is terminated. In these experiments, d_{max} was set to 0.2, $L = 3$ and $T = 2$.

4.4 Incorporating scene context through horizon detection

A common feature of maritime surveillance imagery is that the horizon is visible. In this thesis, 'horizon' refers to the *true horizon*, a theoretical circle defined by the tangential intersection of an observer's line of sight with the surface of the Earth.

In open sea, the horizon corresponds to the boundary where the sea meets the sky (sometimes referred to as the 'sea-sky line'). Land and other features may appear on this boundary, but at large distances they are negligible. When facing the shore or in constrained environments, such as a harbour, the water edge is interrupted by many features in the background (boats, jetties, headlands, etc.) so the true horizon is more difficult to estimate.

Both the horizon line and water edge provide information about the scene which can be further exploited in the object detection process by introducing strong cues / constraints on the size and location of targets. For example, the position of the water edge and/or horizon line can be used to:

- estimate the range of the object, and thereby its horizontal position and size
- discard false positive detections based on their size and location in the image
- infer the instantaneous orientation of the camera, which is needed for projecting detections to real-world coordinates
- stabilise images to compensate for camera roll and pitch
- build a more structured scene model for background subtraction or other methods

4.4.1 Horizon detection

The horizon appears as a straight line in an observer's field of vision or image, provided there are no optical distortions (e.g. caused by camera lenses, observer wearing eye-glasses, windows causing refraction, etc.). At high altitudes (greater that 20,000 feet), it is possible to observe the curvature of the horizon, provided the horizontal field of view is sufficiently large [143]. In this work, the observation heights are well below this threshold so the straight line model is appropriate. However, due to imaging optics, the horizon does often appear as a curve, especially if it is near the top edge of the image. This must be corrected through calibration of the camera and rectification of the image before processing. Horizon detection presents a number of challenges:

- The horizon may not be clearly visible due to low contrast, lighting, weather conditions, etc.
- The horizon may not be visible *at all* in an image (e.g. because the camera is angled too far down or up)
- Occlusion of sections of the horizon by ships or land
- False positives from: edges of ships or land, edges generated by wake (especially if objects are travelling across the field of view), and boundaries created by bands of different coloured water regions

There are many efforts which have addressed the horizon detection problem in air [64–66, 86, 155], land [5, 56, 134, 152] and sea [8, 35, 72, 81, 119, 165, 166, 207, 220]. They can be divided into the following broad categories:

- **Edge-based:** edges are extracted from the image using Canny or similar edge detector and then a procedure is used to extract the longest, straightest, continuous line (which is taken as the horizon) [8, 35, 56, 64, 86, 207].
- **Gradient-based:** the vertical intensity gradient is analysed and a straight line is fitted (e.g. using RANSAC) to the salient gradient points in each column [81, 119, 134, 220].
- **Region-based:** some criteria is used to find the straight line which maximally separates the regions above and below the horizon in terms of their visual appearance [65, 66, 72, 152, 155].

Outside of these categories, there are hybrid methods using both edges and gradients [165, 166] and machine learning methods which use a pixel-wise horizon classifier [5, 106].

The separation score method was trialled but was not found to be a reliable way of discriminating the regions. In many scenes, the optimisation found a line which did not represent the horizon, even when the initial estimate is close to the true position. This occurred with hazy scenes where there is not much contrast between sea and sky regions, and when objects and wake were present in the sea region (see Fig. 4.21). With HD images, the method is also too slow to be used in a real-time system.

However, it would be useful to know if a horizon estimated from a faster method was accurate. This could be done by monitoring the region separation score to see if there is a significant deviation, indicating a possible false detection. However, it was found that this is not possible in practice, as the separation score is affected by objects in the scene.



Fig. 4.21 Examples of horizon detection failure using the region separation method. The white line indicates the initial horizon line guess and the green line is the optimal line found by the algorithm.

In Fig. 4.22, it can be seen that, as the boats approach the camera, the separation score decreases. This is *not* because the horizon line is incorrect, but because the properties (mean and covariance) of the lower region have changed dramatically due to the presence of the boats and their wake.

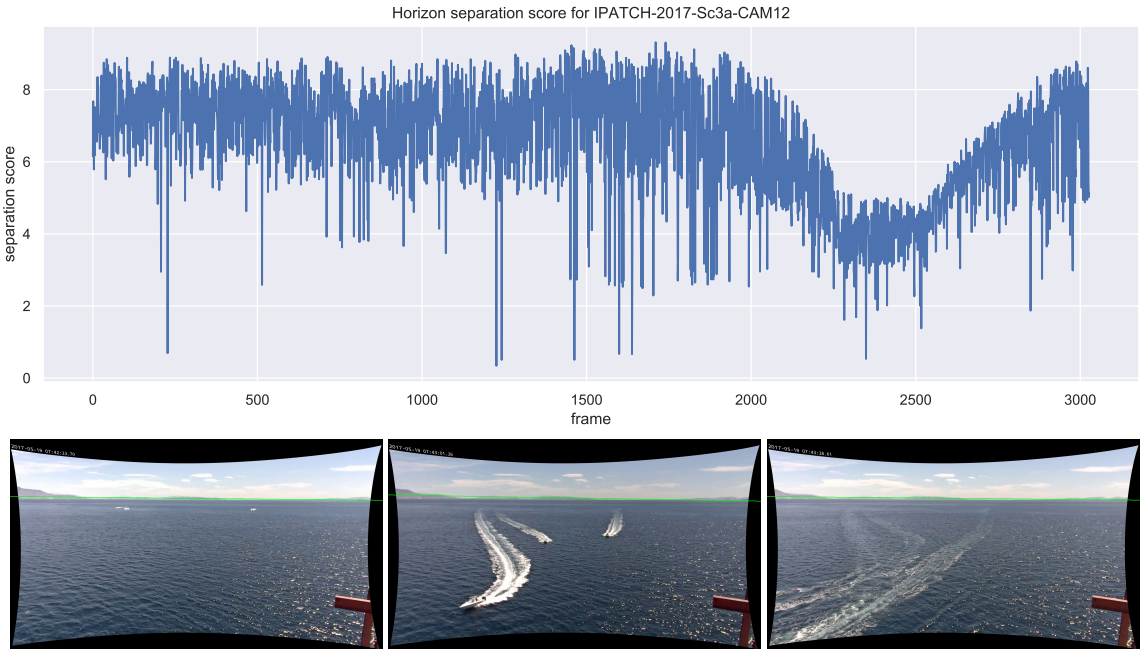
Horizon detection datasets

The majority of papers use their own dataset. Four datasets are available in the maritime domain which provide horizon groundtruth: Buoy [72], MarDCT [27], MODD [119] and SMD [168]. These have been used in the literature [72, 165, 166, 198, 207] for evaluating and comparing horizon detection methods. Other horizon detection works use their own datasets which have unfortunately not been made public.

Horizon detection evaluation

Many papers only provide qualitative evaluation on the public datasets. Some papers provide results (e.g. '99% accuracy') but do not report the criteria used to determine true positives. The most common evaluation measure looks at the statistics of absolute errors in horizon position and orientation (e.g. median, quartiles and histogram) [81, 165, 166, 207]. Other evaluation measures include a pixel-wise accuracy score which measures how correctly the image has been divided in two by the horizon [72] and a true positive evaluation based on whether the horizon position and orientation lie within defined error thresholds [207].

In [165], scores are computed over all sequences aggregated together for each dataset and sub-dataset (SMD is broken down into on-board, on-shore and NIR sub-sets). There



(a) Frame 1800 (b) Frame 2450 (c) Frame 3000

Fig. 4.22 Behaviour of region separation score based on horizons detected by the Park method [158]. From approximately frame 1800 onwards, the boats approach the vessel from a large distance. As they get closer (up to frame 2500), the separation score drops, even though the horizon is correct. The boats exit the scene towards the end of the sequence, causing the horizon score to return to its original level.

is no sequence-level evaluation available. In [166] and [198], sequence level scores are provided, but the sequences are not identified (they are referred to by a number rather than the name of the sequence video file in the dataset). This makes comparison with these results difficult.

The aggregation approach is therefore selected. However, it should also be noted that different numbers of sequences and different frames per sequence are reported than are actually present in the dataset and it is not possible to know which ones were used. Therefore, this comparison is not ideal, but the overall statistics should give some indication of whether performance is similar or not.

The MODD dataset provides groundtruth for the water edge, but not the horizon, as the purpose is to evaluate segmentation of the sea, shore and sky regions. The groundtruth is provided in multiple discontinuous sections due to the occluding objects and land in the scene. As explained earlier, this makes it unsuitable for use as a horizon to evaluate position and orientation errors. Additionally, there are no numerical results reported in the literature for the MODD water edge detection (other than in the original work [119]), although some qualitative results are reported in [207]. For these reasons, the MODD horizon data is not included in this analysis.

Selected method

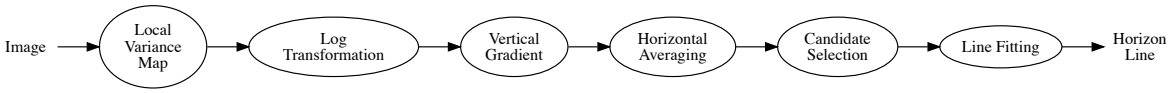


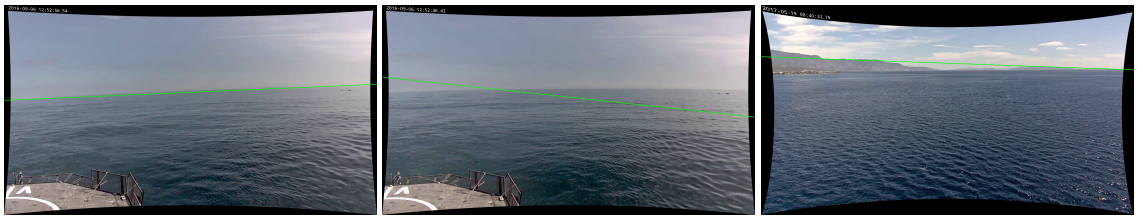
Fig. 4.23 Selected horizon line detection method from Park et al. [158]

The vertical gradient-based method from Park et al. [158] (Fig. 4.23) is adopted as the horizon detector, based on its simplicity, speed and performance. The method first creates a local variance map, V, and log-transforms it (L) to enhance regions of smaller change.

$$V(x,y) = \sum_{i}^{m \times m} \left(p_i - \bar{p}(x,y)\right)^2 \tag{4.23}$$

where p_i is a pixel in the $m \times m$ neighbourhood centered on $p(x,y)$, and $\bar{p}(x,y)$ is the neighbourhood mean.

$$L(x,y) = \log\left(V(x,y) + 1\right) \tag{4.24}$$



(a) (b) (c)

Fig. 4.24 Example horizon detection results on IPATCH sequences: (a) Detection is possible even with low contrast horizons; (b) False detections still occur but can be mitigated through smoothing over time; (c) The presence of a strong edge from land can cause incorrect detections.

The 1 is added to avoid log(0). A vertical gradient map, G, is computed through simple differencing in the vertical direction and averaging is performed per row within vertical segments to form an image of bars, B.

$$G(x,y) = \left| L(x,y) - L(x,y-1) \right| \tag{4.25}$$

$$B(k,y) = \frac{1}{P} \sum_{x=kP}^{(k+1)P-1} G(x,y), \text{ for } k = 0,1,\ldots,K-1 \tag{4.26}$$

where P is the width of the column segment and K is the number of column segments.

Finally, the parameters of the horizon line are estimated by fitting a straight line to the coordinates of the centre of each bar in B using linear least squares regression.

Evaluation

The selected horizon detector method was implemented and compared against results reported for other methods from the literature (Tables 4.5a, 4.5a and 4.6, and Fig. 4.24). In addition to varying the size of the column segments, downsampling of the image was applied to increase speed. Three parameter sets are used:

1. Downsample factor: 1, columns: 120 ('Park-d1-c120')
2. Downsample factor: 2, columns: 120 ('Park-d2-c120')
3. Downsample factor: 4, columns: 60 ('Park-d4-c60')

Table 4.5 Results for horizon detection methods on different datasets (best and second best highlighted in green and blue, respectively). Values for the Park methods obtained in this work; other values as reported in [165, 166, 168, 207].

(a) Median absolute position error (lower is better)

Method	Buoy	Mar-DCT	Onboard	Onshore
ENIW [65, 66]	1.93	37.43	117.81	115.25
FGSL [72]	1.59	198.58	118.14	115.25
GWR [8]	2.1	223.41	436.88	42.24
Hough [81]	1.92	198.15	221.02	206.40
IntG [81]	11.97	271.91	340.20	5.00
IntGF [81]	6.00	122.59	351.53	178.29
MSCM-LiFe [166]	-	-	3.2 - 6.5	1.9 - 8.9
MuSCoWERT [165]	**1.44**	**1.33**	**1.49**	**2.63**
MuSMF [35]	3.98	162.43	279.74	66.48
Radon [84, 168]	-	-	362	359
Park-d1-c120	**1.08**	**2.46**	2.33	4.24
Park-d2-c120	1.88	4.11	**1.91**	**3.81**
Park-d4-60	3.83	6.61	4.11	5.60

(b) Median absolute angle deviation (lower is better)

Method	Buoy	Mar-DCT	Onboard	Onshore
ENIW [65, 66]	0.24	**0.26**	0.47	0.18
FGSL [72]	**0.20**	0.64	0.49	0.18
GWR [8]	0.29	0.57	2.00	0.18
Hough [81]	**0.19**	0.64	0.64	0.19
IntG [81]	1.50	0.64	1.03	**0.14**
IntGF [81]	0.39	2.07	1.04	1.24
MSCM-LiFe [166]	-	-	0.5 - 0.7	0.3 - 0.7
MuSCoWERT [165]	0.27	**0.36**	**0.25**	0.21
MuSMF [35]	0.32	0.99	1.08	0.73
Radon [84, 168]	-	-	3.4	0.4
Park-d1-c120	0.22	0.55	**0.25**	**0.13**
Park-d2-c120	0.30	0.56	**0.19**	0.24
Park-d4-60	0.54	1.13	0.31	0.41

Table 4.6 Processing speed (mean ms/frame) for horizon methods (values reported in [165, 166, 168, 207])

	Buoy 800 × 600	Mar-DCT *various*	SMD 1920 × 1080	Sun et al. [207] 640 × 800
CFS [207]	-	-	-	94
ENIW [65, 66]	~minutes	~hours	~hours	-
FGSL [72]	3,700	9,400	12,800	-
GWR [8]	100	200	400	-
Hough [81]	50	100	300	-
IntGF [81]	30	60	90	-
IntG [81]	20	50	70	-
MSCM-LiFe [166]	-	-	-	231
MuSMF [35]	500	700	900	-
MuSCoWERT [165]	5,800	7,000	9,500	-
Radon [84, 168]	-	-	2,700	-
SSM [119]	-	-	-	27
Wang [220]	-	-	-	57
Park-d1-c120	10	7	42	-
Park-d2-c120	5	5	14	-
Park-d4-60	3	3	6	-

Summary of results

Table 4.5a and 4.5a show that the selected method is among the top performing methods across all datasets. Its speed, however, is far superior to other methods (Table 4.6) so is a good choice for a real-time application. Fig. 4.24 shows some example detections from the Park method on IPATCH sequences. Some false detections still occur but these can be mitigated through averaging over frames. Further improvements could be gained by restricting the horizon search window so that false detections from other regions are reduced. The results also indicate that downsampling improves speed without harming performance and in some cases, actually improves it. The Park-d2-c120 is therefore adopted throughout the rest of this chapter and in Chapter 6 as the horizon detection method.

4.4.2 Depth-weighted activation maps in BMS

Because of the large viewing range in maritime scenes, distant objects and regions of the scene are mapped to a small number of pixels. This has the effect of blurring pixel intensity values. This is further compounded by atmospheric haze, which causes colour definition to be reduced, and image compression, which exploits low contrast regions to discard information.

Small objects are still *locally* salient under the BMS surroundedness framework, but they are not *globally* salient compared to closer objects which benefit from sharp resolution and contrast in the image. Applying a global saliency threshold as per the previous section tends to *over*-detect sparkle and glint which is close to the camera, but *under*-detect distant objects of similar size.

Another problem is in the formulation of BMS itself. In order to promote activation maps with a few, small regions over those with large regions, L2 normalisation is used (Eqn. 4.14). Because activation maps are binary, this is equivalent to

$$w_i = \frac{1}{no.\ of\ non\text{-}zero\ pixels} \tag{4.27}$$

meaning that each map gets a weight which is inversely proportional to the number of activated pixels.

This is fine, but the weight is applied equally to every pixel in the map *without taking into account its context.* If a small, distant target is activated in the same map as lots

of other small regions caused by noise or glint, it will get a lower weight than if it was activated on its own. This makes it difficult to detect small distant objects under the current BMS implementation.

The proposed solution to this is to modify the BMS method using a map which encodes depth of the scene. Instead of weighting each activation map using L2 normalisation, the proposed solution weights each activation map in a way that is location dependent. This means that surrounded regions near the horizon are given more emphasis than those nearer the vessel. The idea of modifying an image using horizon regions has been used in a similar way in [118] for infrared images and small targets. The activation map normalisation step in BMS (Eqn. 4.14) is modified as follows:

$$\bar{A}_i = w_d(x,y)\left(\mathcal{N}\left[A_i^+ \oplus K_{D1}\right] + \mathcal{N}\left[A_i^- \oplus K_{D1}\right]\right) \tag{4.28}$$

where $w_d(x,y)$ is a pixel-wise weight map based on inferred depth in the scene and $\mathcal{N}[.]$ is a normalisation function which maps to the range $[0,1]$. As a further filtering step, detected salient regions can be excluded if they are wholly[7] above the horizon to reduce false positives. Fig. 4.25 shows the effect of weighting the activation maps in BMS.

4.4.3 Creating a depth map from the horizon

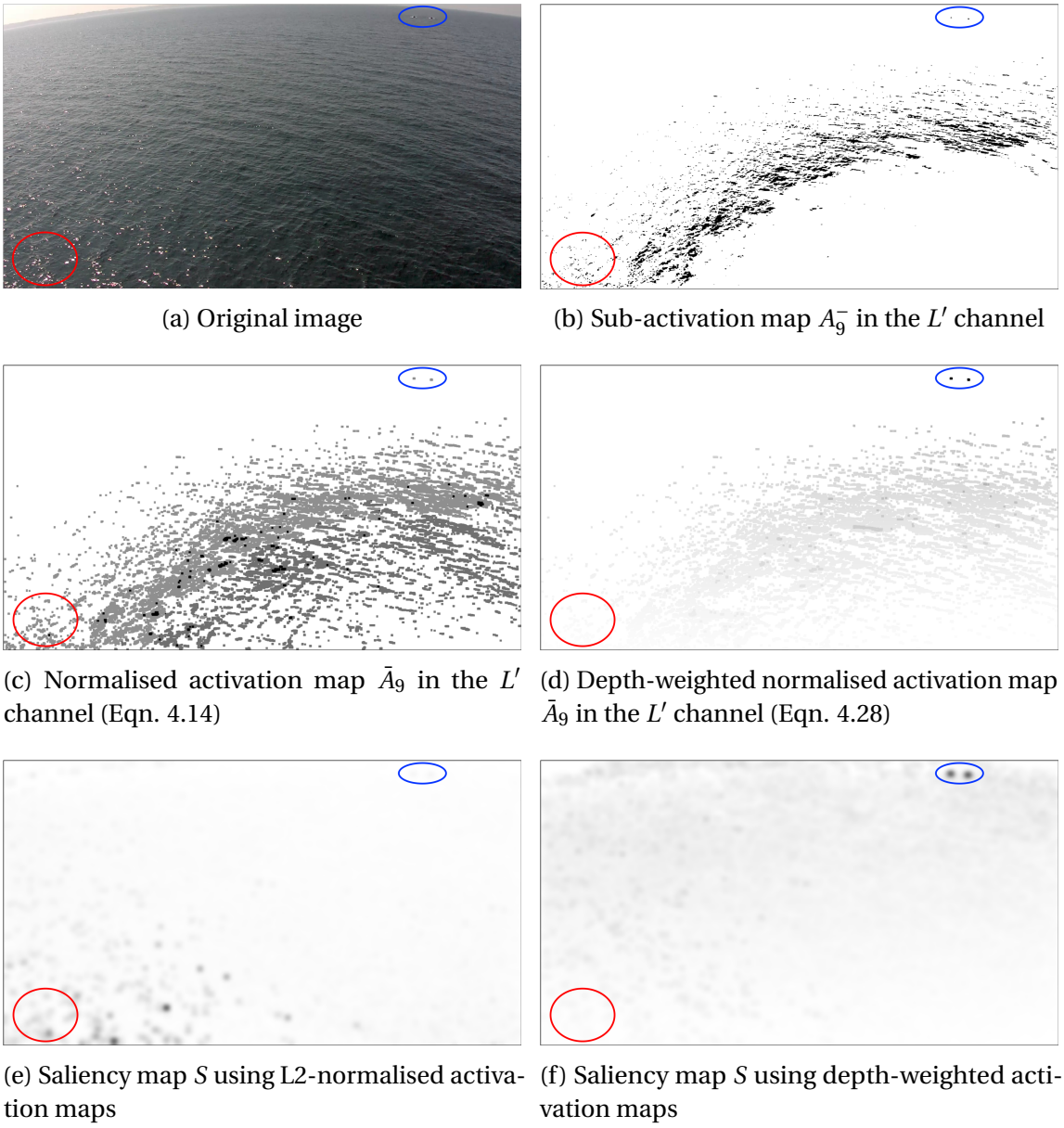
The weighting map could be manually 'hard coded' but in this work, the position of the horizon is used to create a map which represents depth in the scene and can change as the viewpoint moves. The theoretical distance to the horizon (Fig. 4.26a), H_0, depends only on the observer's height:

$$H_0 = \sqrt{2Rh + h^2} \tag{4.29}$$

where R is the radius of the Earth and h is the height of the observer above the Earth's surface. This model assumes a perfectly spherical Earth with $R = 6,371$km taken as the average radius. It also assumes no refraction from the atmosphere, which can extend the apparent distance to the horizon.

Following [130], if the position of the horizon is visible to the observer, the relative distance to a point on the surface of the sea can be derived from the vertical angle subtended between the horizon and that point. Fig. 4.26b and 4.26c shows this set-up.

[7] Note that objects can overlap the horizon, so the system can only confidently exclude objects which are entirely above the horizon



(a) Original image

(b) Sub-activation map A_9^- in the L' channel

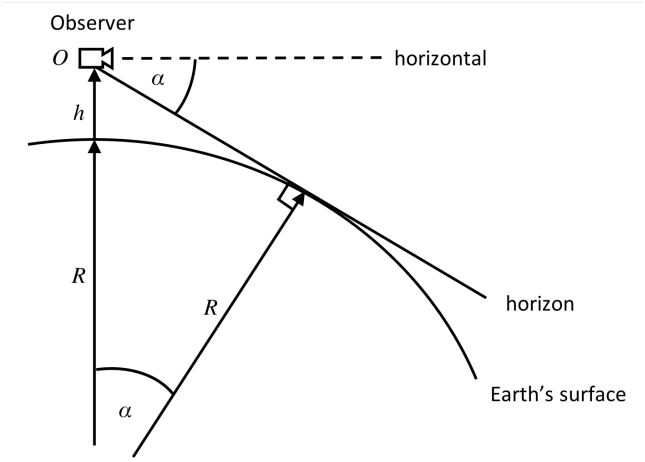
(c) Normalised activation map $\bar{A}_9$ in the L' channel (Eqn. 4.14)

(d) Depth-weighted normalised activation map $\bar{A}_9$ in the L' channel (Eqn. 4.28)

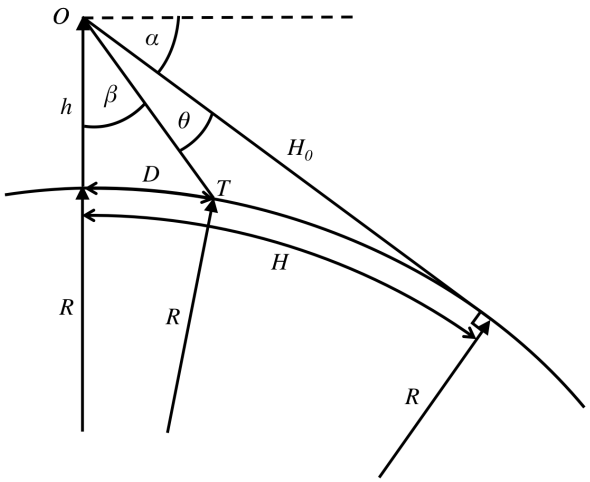
(e) Saliency map S using L2-normalised activation maps

(f) Saliency map S using depth-weighted activation maps

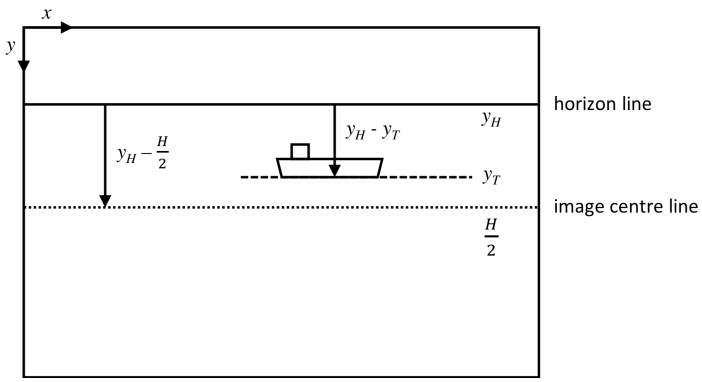
Fig. 4.25 Effect of using scene depth to weight the activation maps in BMS. Blue circle highlights position of distant boats, red circle shows sparkle close to vessel. The surrounded regions of the boats and sparkle are of similar size in the sub-activation map (b). Using L2 normalisation (c), the boats and sparkle are given equal weight in each activation map and sparkle becomes more salient overall (e). Using the proposed depth weighting approach (d), more distant surrounded regions are given more weight and remain prominent in the final saliency map whilst nearby sparkle is suppressed (f).



(a)



(b)



(c)

Fig. 4.26 Horizon geometry

To calculate the distance from the observer to the target T, the angle β must be calculated:

$$\beta = \frac{\pi}{2} - \alpha - \theta \tag{4.30}$$

Angle α is the angular drop from the horizontal to the horizon and θ is the angle subtended between the horizon and the target:

$$\alpha = \tan^{-1}\left(\frac{H_0}{R}\right) \tag{4.31}$$

$$\theta = \tan^{-1}\left(\frac{y_H - y_T}{f_y}\right) \tag{4.32}$$

where y_H and y_T are the y-coordinates of the horizon and target in the image (Fig. 4.26c), and f_y is the focal length of the camera in the y-axis.

Finally, the distance from the base of the observer to a target point can be computed using Equation 4.33:

$$D = (R+h)\cos\beta - \sqrt{(R+h)^2\cos^2\beta - \left(2Rh + h^2\right)} \tag{4.33}$$

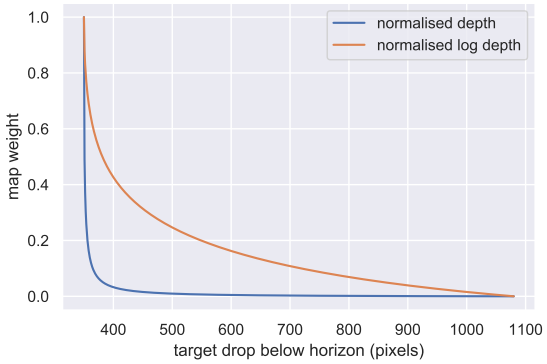
In addition, the angle of camera pitch (β_c) and roll (α_c) can be determined from the position and orientation of the horizon, respectively:

$$\begin{aligned} \beta_c &= \delta - \alpha \\ \delta &= \tan^{-1}\left(\frac{y_H - \frac{H}{2}}{f_y}\right) \end{aligned} \tag{4.34}$$

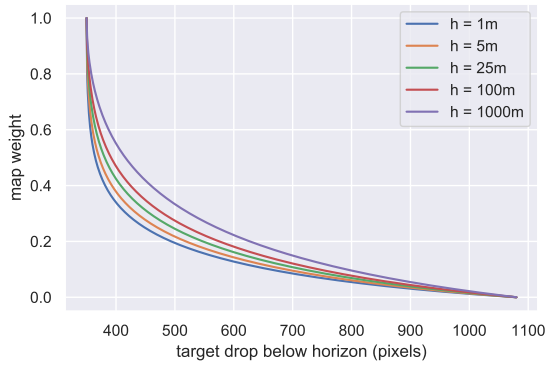
$$\alpha_c = \tan\left(-\phi\right) \tag{4.35}$$

where H is the image height in pixels, δ represents the angle subtended between the horizon line and the midline of the image and ϕ is the orientation of the horizon line (measured from the x-axis). These values can be useful for estimating or validating the orientation of the camera (and hence vessel) in each frame.

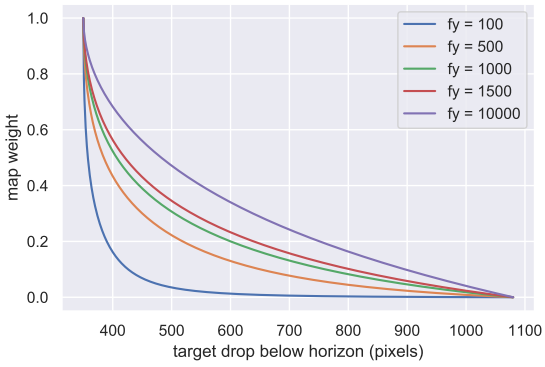
Note that knowledge of the camera – its height h and focal length f_y – is required to compute the depth map. Fig. 4.27b shows that the depth map is not very sensitive to small changes in observer height, so this can be estimated if precise measurements are not possible. This is useful for the case when the system is mounted on a ship, as motion of the waves causes changes in the height of the camera. The depth map is more



(a) Horizon depth map functions



(b) Sensitivity of depth map to observer height



(c) Sensitivity of depth map to focal length

Fig. 4.27 Horizon-based depth map functions

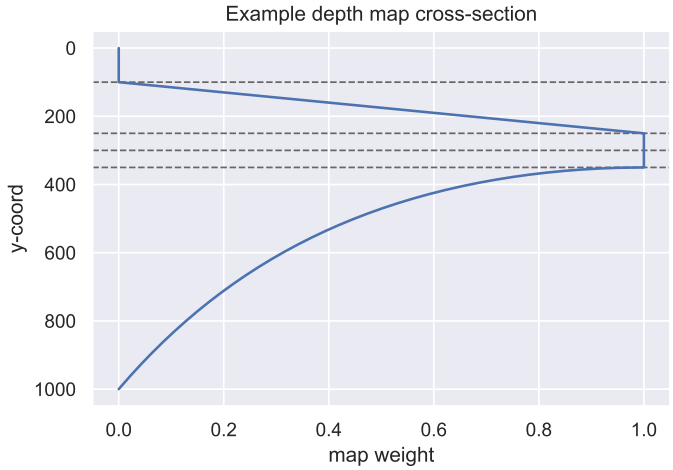


Fig. 4.28 Example depth map cross section

sensitive to focal length (Fig. 4.27c), particularly at lower values, so care should be taken when calibrating this parameter. The cameras in the IPATCH system were calibrated and found to have a focal length of around 1300. Small calibration errors at this range are not significant.

The depth weighting map is calculated as a function of pixel distance in the image *perpendicular to the horizon* to account for roll of the camera. In the region below the horizon, the depth formula, D (4.33), is used. To manage uncertainty around the horizon, a fixed width bar is established with maximum weight (= 1.0) spanning from w_2 above the horizon to w_3 below. Above the horizon, linear decay is applied for a fixed width (w_1), above which the map weight is set to 0. Mathematically, the depth map is constructed in four different sections, as follows:

$$w_i = \begin{cases} 0 & \text{if } 0 \leq y < y_H - w_2 - w_1, \\ 1 - \frac{y_H - y - w_2}{w_1} & \text{if } y_H - w_2 - w_1 \leq y < y_H - w_2, \\ 1 & \text{if } y_H - w_2 \leq y < y_H + w_3, \\ \mathcal{N}\left[\log\left(D\left(y\right)\right)\right] & \text{otherwise.} \end{cases} \tag{4.36}$$

where $\mathcal{N}\left[.\right]$ is a normalisation function which maps to the range $[0,1]$. The log transformation is used to compress the large distance range and represent *relative* distance (Fig. 4.27a). Fig. 4.28 shows the cross section of an example depth map with $y_H = 300$, $w_1 = 150$, $w_2 = 50$ and $w_3 = 50$.

4.5 Evaluation and comparison against baselines

4.5.1 Experimental set-up

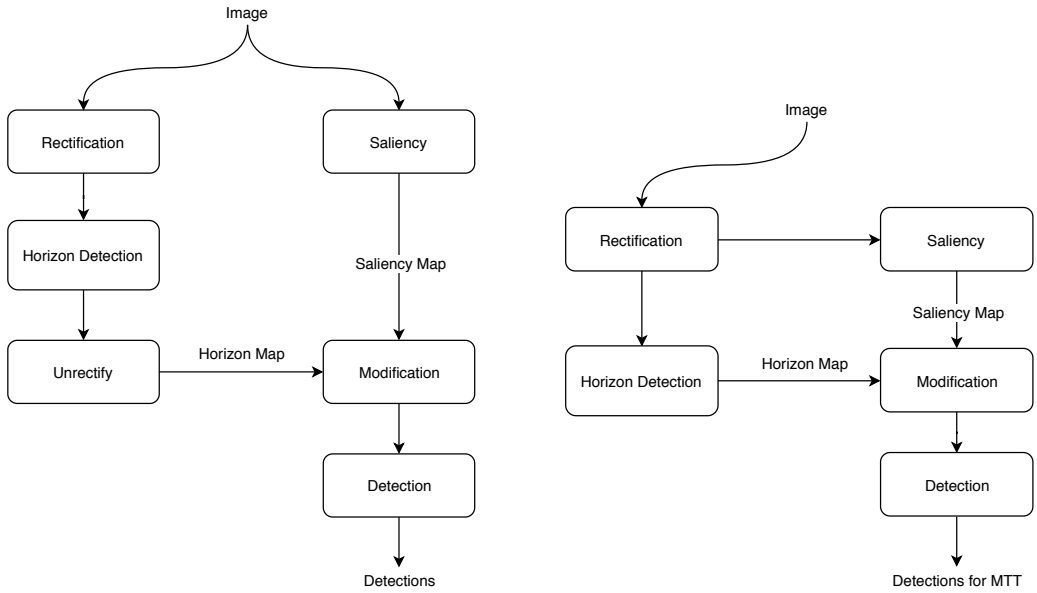
Sequences and metrics

The sequences used for evaluation are listed in Table 3.3. The horizon is not visible in the SEAGULL sequences and the camera calibration is not known for the SMD data, so the horizon depth map step is only evaluated on the IPATCH sequences. This will be further exploited in the on-board system testing in Chapter 6. The MODP-BEP3, Detection Rate and FAF metrics are used for quantitative analysis, as described in Chapter 3. In addition, to assess real-time performance, the processing speed of the proposed method is measured for each frame.

Implementation and configurations

The BMS algorithm parameters were set to the values reported in [241] ($\delta = 8$, $D1 = 7$, $D2 = 9$ and $\sigma = 9$). Thresholds at the 99.5^{th} and 99th percentiles are taken as the baseline. The size of horizontal and vertical strips are set to 100 pixels. Temporal filtering is configured as described in Section 4.3.4. For the IPATCH sequences, horizon detection was performed in the rectified image and the generated horizon map was 'unrectified' back to the original image space for alignment with the saliency map (Fig. 4.29a). This was necessary as the image groundtruth has been prepared in the *unrectified* image. In the Chapter 6, the real-world surveillance system needs detections from the rectified image so this step is unnecessary (Fig. 4.29b).

The saliency and horizon detection methods were implemented in C++ and run on a MacBook Pro with 2.6GHz Intel® Core™ i7 processor and 16GB RAM. Table 4.7 summarises the variants of the saliency method which were used in the experiments.



(a) This chapter (b) Chapter 6

Fig. 4.29 Integration of horizon detection in the saliency method. Horizon detection is performed in the rectified image. In this chapter, the horizon map is 'unrectified' back to the original image space because evaluation is performed using groundtruth annotations which were created in the original images. In Chapter 6, the multi-target tracker (MTT) needs detections from the rectified image so this process is not necessary.

Table 4.7 Key to saliency method variants

Variant Name	δ	T1, T2	Threshold	Horizon	Downsample
saliency-995-99-d8	8	99.5, 99	Global	-	-
saliency-99-98-d8	8	99, 98	Global	-	-
saliency-99-95-d8	8	99, 95	Global	-	-
saliency-995-99-d8-hv	8	99.5, 99	Hori. Vert.	-	-
saliency-995-99-d8-depth	8	99.5, 99	-	✓	-
saliency-995-99-d8-hv-depth	8	99.5, 99	Hori. Vert.	✓	-
saliency-995-99-d16	16	99.5, 99	Global	-	✓
saliency-995-99-d8-ds	8	99.5, 99	Global	-	✓

4.5.2 Results and analysis

Effect of thresholding values

With larger objects, lower percentile thresholds more successfully capture the whole object. SMD-1615 is a good example of this. With the high thresholds (99.5 and 99), the objects are fragmented (Fig. 4.30a). With lower thresholds (in particular the lower hysteresis threshold), the whole object region is recovered (Fig. 4.30b and 4.31). However, with small objects, lower threshold values are likely to capture more background, especially if the background is highly-salient wake. This is the case in IPATCH-Sc2a_Tk1-CAM11 (Fig. 4.31c), leading to a decrease in performance for lower thresholds.

The same applies for scenes with large numbers of objects (i.e. a greater total target area). The percentile-based approach assumes that the objects only occupy a certain proportion of the image. With larger objects or denser scenes, this assumption breaks down and the thresholds must be lowered accordingly. Conversely, scenes where only one very small object is expected actually benefits from higher thresholds. This is the case for the two SEAGULL sequences (Fig. 4.32a).

Effectiveness of horizontal vertical thresholding vs. global thresholding

Fig. 4.32b and 4.33 show the effect of horizontal-vertical thresholding. Compared to the global thresholding, the horizontal-vertical thresholding approach is able to more accurately extract targets from wake regions, although it can also create additional false detections. Another benefit of the horizontal-vertical thresholding is the enhanced ability to detect distant targets such as those in the IPATCH sequences. However, with larger targets (such as in the SMD sequences), the horizontal-vertical thresholding only captures a small salient point on the object, leading to lower MODP scores.

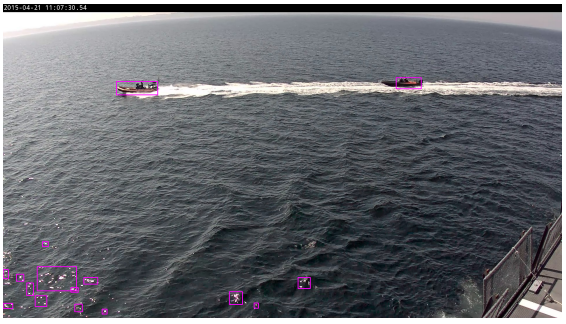
In contrast to the feature map-based methods, the BMS method does not explicitly consider scale. Introducing scale into the method (e.g. through scale space representation) could be a way to reduce the effects observed in Fig. 4.33. Scale space analysis would allow detections to be extracted at multiple scales. Further processing could then be used to select the most appropriate scale to detect objects based on the scene context. However, computing the BMS method over multiple image scales would be computationally expensive.



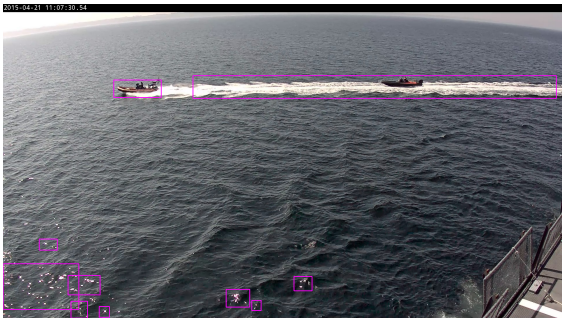
(a) $T_1 = 99.5$, $T_2 = 99$ (image cropped)



(b) $T_1 = 99$, $T_2 = 95$ (image cropped)



(c) $T_1 = 99.5$, $T_2 = 99$



(d) $T_1 = 99$, $T_2 = 95$

Fig. 4.30 Effect of different thresholding values on the SMD-1615 (a-b) and IPATCH 2015-Sc2a_Tk1-CAM11 (c-d) sequences. With a lower and wider threshold range (b and d), more of the image is extracted. This is beneficial for larger objects in SMD, but leads to overestimation of the target in IPATCH.

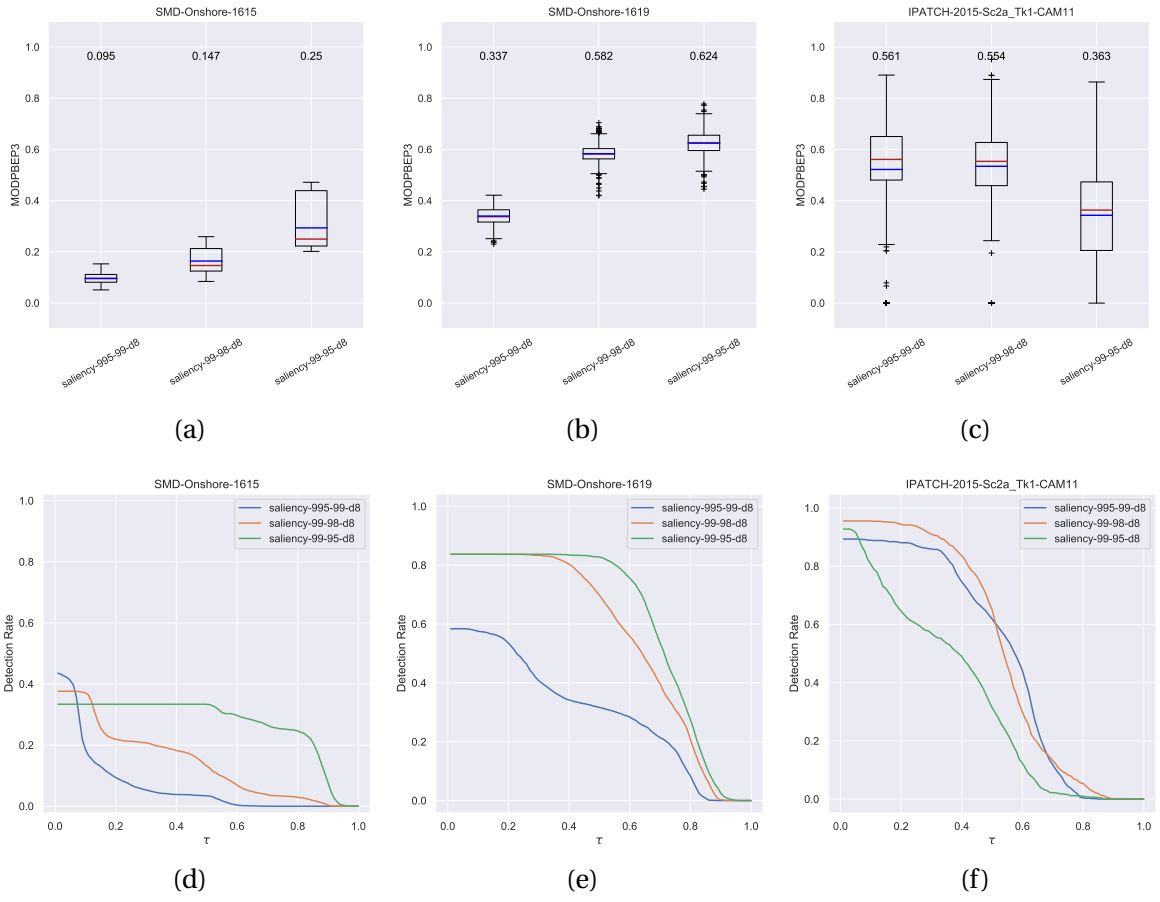
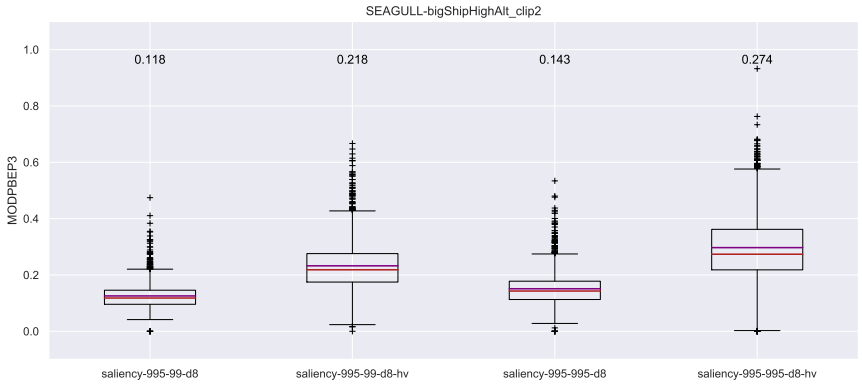
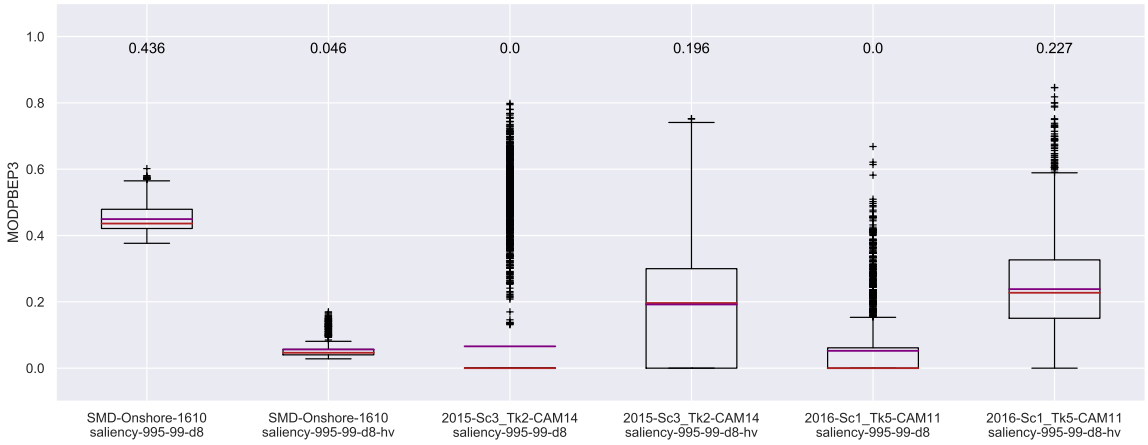


Fig. 4.31 Effect of different thresholding values on MODP-BEP3 for the SMD-1615 and IPATCH 2015-Sc2a_Tk1-CAM11 sequences

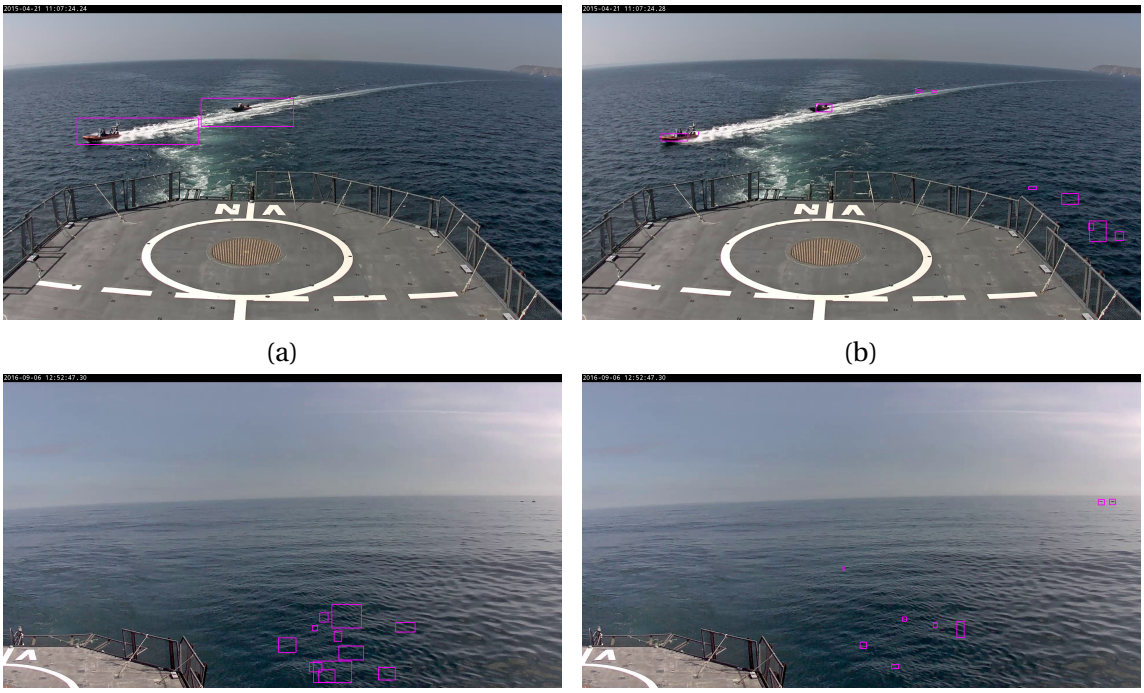


(a) Effect of high threshold values and horizontal-vertical thresholding on SEAGULL sequence



(b)

Fig. 4.32 Effect of horizontal-vertical thresholding on SMD and IPATCH sequences



(a) (b)

(c) (d)

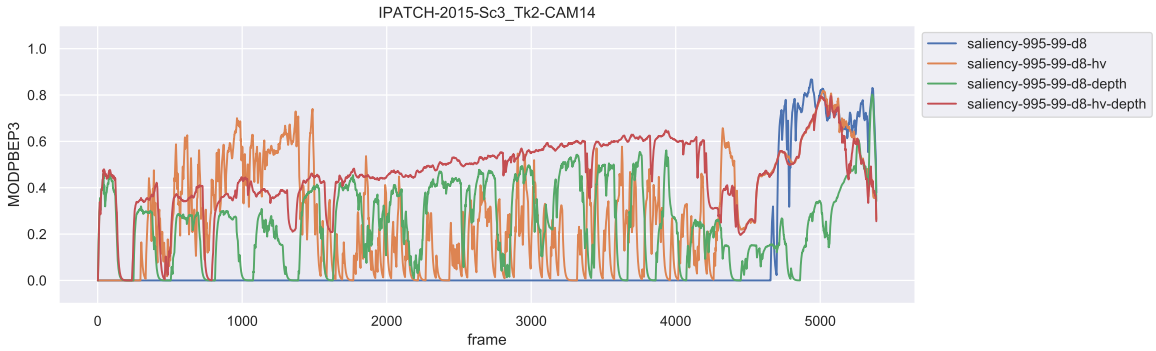
Fig. 4.33 Effect of horizontal-vertical thresholding on wake and reflections

Effectiveness of depth map weighting

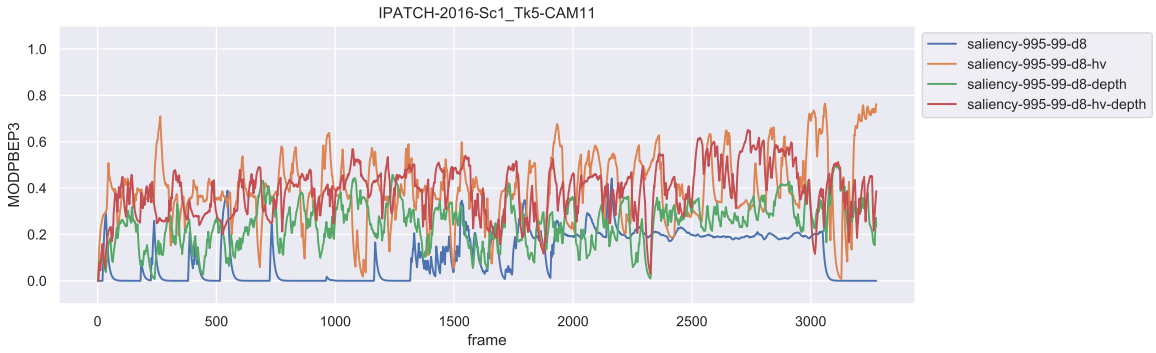
Fig. 4.34 looks at the effect of depth map weighting applied to an IPATCH sequence from each of the 3 campaigns in which the skiffs approach the vessel from the distance. Without horizontal-vertical thresholding or depth map weighting, the saliency method only detects the targets when they get close/large enough (blue lines in Fig. 4.34). With horizontal-vertical thresholding and/or depth map weighting applied, the distant targets can be detected much earlier. However, objects closer to the camera can be detected less well as a result, for example in Fig. 4.34a and 4.34c the blue line overtakes the others towards the end of the sequence as the objects get closer. Note the cyclical patterns that occur in Fig. 4.34a and 4.34b due to the motion of the vessel.

Temporal filtering and false positives

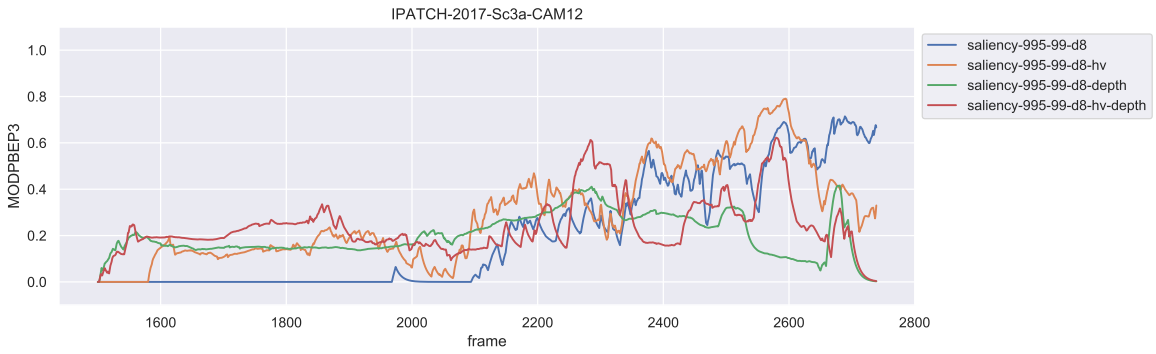
Looking at the same three IPATCH sequences, Fig. 4.35a shows that the proposed horizontal-vertical thresholding and depth map weighting steps reduce the number of false positives compared to the baseline, although the level is sequence dependent. The temporal filtering step is having the desired effect of reducing the number of false positives for all



(a) IPATCH 2015-Sc3_Tk2-CAM14



(b) IPATCH 2016 Sc1_Tk5-CAM11



(c) IPATCH 2017-Sc3a-CAM12

Fig. 4.34 MODP-BEP3 vs. frame number for IPATCH sequences where targets approach from a large distance (MODP-BEP3 values are smoothed to reduce noise and show trends more clearly).

configurations and all sequences without adversely affecting the distribution of MODP-BEP3 scores (Fig. 4.35b). In fact, in most cases, the filtering process improves localisation accuracy.

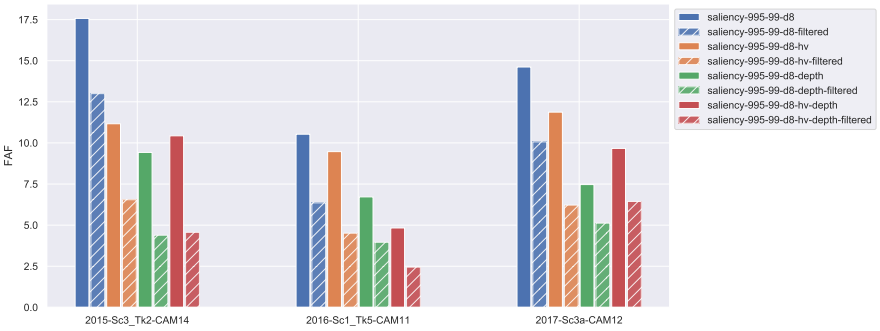
Trading off speed and performance

Whilst the BMS method is fast on salient object detection benchmarks, its speed on larger images (such as those in the IPATCH data) is not sufficient for real-time operations. The authors of BMS report in [240] that they did not see a significant drop in performance (AUC score) when they increased the sampling step size (δ) to create fewer Boolean maps per image. This was tested to see if it would be the case also for the IPATCH data by setting $\delta = 16$. Performance and speed were also tested on downsampled versions of the images (factor 2) for comparison (Fig. 4.36). Changing the step size δ gave a small gain in speed with a small drop in performance. Downsampling the image gave a more significant increase in speed but a much bigger drop in performance.

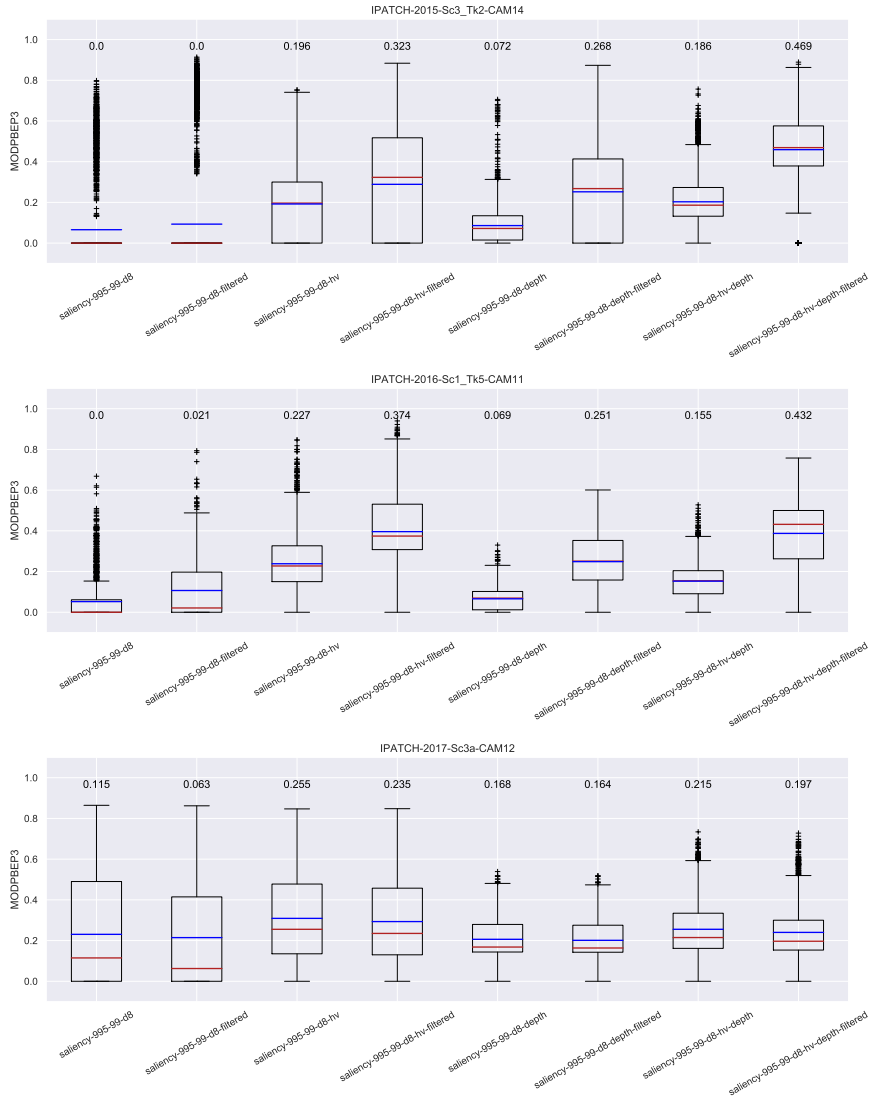
Limitations

For real-time systems, a key weakness of the proposed saliency-based approach is the time required to process all the different maps in the BMS step. Some speed could be gained from optimising the code implementation (e.g. parallelisation) and using larger values of δ. Downsampling the image is the most effective way to increase speed but this means that distant targets will not be detected. Another limitation is the tendency to merge two objects if they are close or overlapping. This is caused by the post-processing step in BMS which applies blurring to reduce noise in the final saliency map. This step can be turned off, but this leads to a lot of false detections and fragmentation of objects.

The saliency approach is based on an assumption that the objects of interest will be the most salient things in the scene. If this is not the case, for example if there is land or many other objects in the scene, this assumption breaks down. This leads to objects being missed or fragmented (if threshold are set high) or objects being over-detected (if thresholds are set low). Finally, wake and reflections continue to create false positives, although these are mitigated to some extent by the proposed horizontal-vertical thresholding, depth map weighting and temporal filtering steps. The human visual system, whilst initially drawn to these features and regions, performs additional (top-down) processing to distinguish

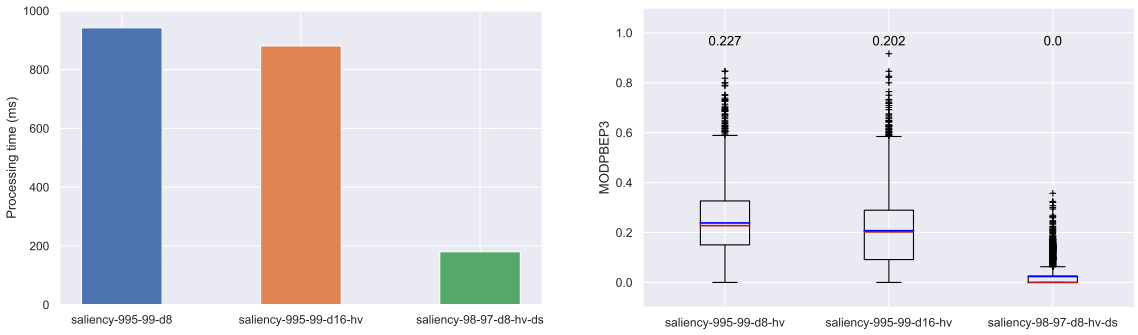


(a)



(b)

Fig. 4.35 Effect of temporal filtering on false positives (a) and MODP-BEP3 performance (b) for IPATCH sequences where targets approach from a large distance.



(a) Average processing time per frame for different

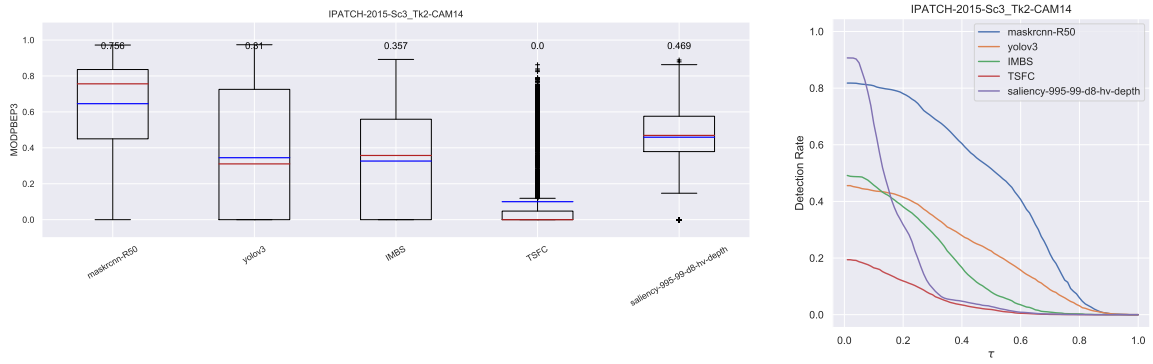
(b) MODP-BEP3 performance for IPATCH 2016-Sc1_Tk5-CAM11

Fig. 4.36 Speed vs. performance trade-off

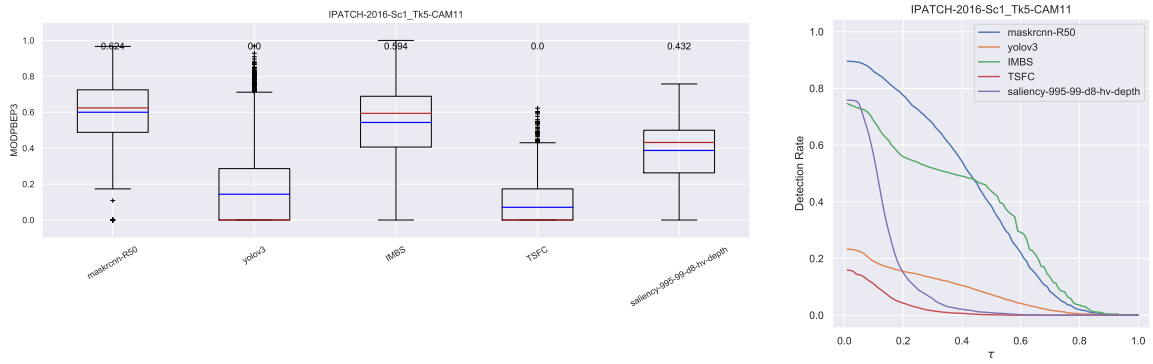
between the different objects (wake, boat, etc.). The addition of top-down processing could improve this further.

Comparison against baselines

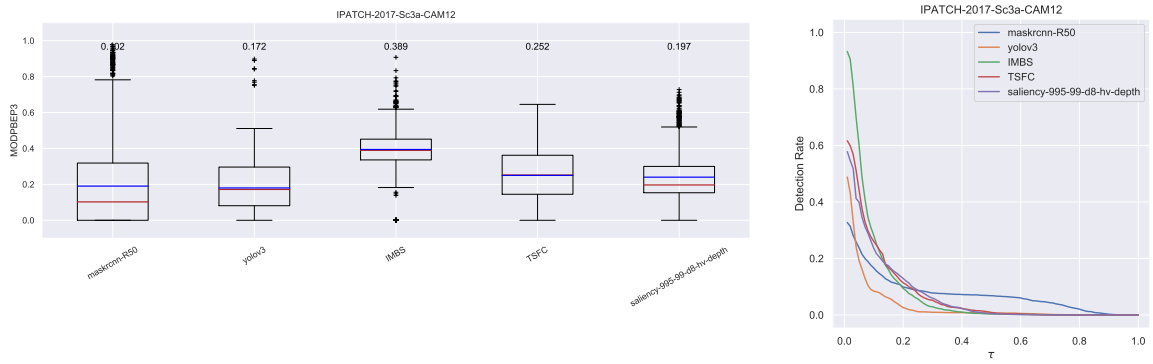
Higher thresholds and the proposed thresholding and depth map weighting steps favour sparse scenes with smaller, distant targets over busy scenes with larger objects. As the former is more representative of the piracy detection case (detecting as early as possible), the 'saliency-995-99-d8-hv-depth' is selected as the most appropriate for addressing the challenge of detecting pirate skiffs approaching from a distance. See Fig. 4.37 for results on representative IPATCH sequences. Looking at the MODP-BEP3 distributions, the proposed saliency method achieves moderate performance on all the sequences but is less sequence-dependent (i.e. it is more consistent over the sequences). For early detection of piracy threats, a key factor is detection rate, regardless of localisation accuracy (as location can be refined as the objects get closer). The proposed saliency method has high detection rate curves at lower thresholds across all sequences which is good for initial detection. The detection rate falls quite sharply compared to other methods, indicating that it is not such a good choice for accurately localising the object in the image.



(a) IPATCH 2015-Sc3_Tk2-CAM14



(b) IPATCH 2016-Sc1_Tk5-CAM11



(c) IPATCH 2017-Sc3a-CAM12

Fig. 4.37 Comparison of the proposed saliency-based object detection method with baseline methods from the literature.

4.6 Summary

In this chapter, the concept of using visual saliency as a basis for object detection was investigated. Evaluation of a range of visual attention and saliency methods on maritime surveillance data indicated that BMS [240] was a good candidate. To perform object detection, an adaptive hysteresis thresholding method was proposed, however a lot of wake was still detected and distant objects were often missed.

To address the first issue, a horizontal-vertical thresholding step was proposed which emphasises local saliency, rather than global. The second issue was addressed by incorporating scene context into the saliency step by weighting activation maps in BMS with a map that encodes depth in the scene. The map was generated by detecting the horizon and estimating a depth function through knowledge of the camera height and focal length. The Park horizon detection method [158] was selected from the literature as it was an efficient method, as well as being among the top performing. Finally, temporal filtering was used to smooth the positions of the detections over time and filter out transient salient regions. Not only did this reduce the number of false positives, but it improved the localisation accuracy in most sequences.

The proposed approach exploits the fact that maritime objects are relatively rare in most surveillance contexts (compared to real-world general object detection). This is valid for the piracy detection case but does not hold for the busy port vessel traffic monitoring case. One advantage of saliency-based approaches is that they do not make any assumptions about object size or appearance and generalise to scenes with different viewpoints, backgrounds and conditions. However, the system is deprived of key information which could be used to improve performance, namely the fact that the system operates over maritime scenes which have certain characteristics. In the next chapter, machine learning is applied to create a model of the elements that make up maritime scenes to perform more context-specific reasoning about the presence and location of objects.

Chapter 5

Semantic Segmentation for Object Detection

5.1 Introduction

The state of the art in object detection is represented by deep convolutional neural networks which output bounding boxes and associated class labels. However, as shown in Chapter 3, when applied directly to maritime surveillance data, their performance is more variable. This is because there is a significant gap between the training and runtime (test) data domains. An obvious solution to this would be to train on images which are more specific to the maritime surveillance domain. Unfortunately, labelled training data in this area is limited and expensive to produce.

In another area of research, deep networks are achieving high performance in semantic segmentation and scene parsing tasks such as autonomous driving and augmented reality [41, 52, 79]. Here, the number of images required for training has been observed to be much less (100s of images) [13], which makes it attractive for the maritime domain. However, maritime scene segmentation data is practically non-existent and even more expensive to produce. Driven by the potential application in autonomous vehicle navigation, many of the architectures [147, 159] are also designed with speed and memory consumption in mind so that they can run in real time on low-power hardware, making them suitable for real-world, real-time applications.

The idea explored in this chapter is, rather than learning to find maritime objects, instead learn to segment the whole scene into the most common components of maritime

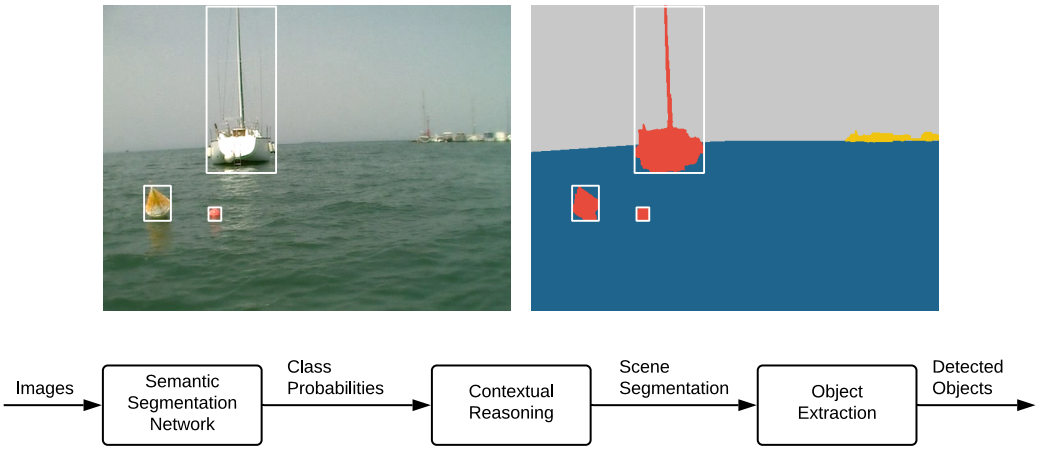


Fig. 5.1 The semantic segmentation-based object detection concept. A semantic segmentation network predicts the class of each pixel as 'Sea' (blue), 'Sky' (grey) or 'Other'. Contextual reasoning is then used to distinguish target objects (red) from other features (yellow), such as land.

environments – sea, sky and (in some cases) land. The object detection problem can then be reframed as the task of finding things which are "not sea or sky". Of course, some training data is still required, so publicly available scene segmentation datasets are used. These are still quite a long way from the test data domain, so the key research questions in this chapter are:

1. Does the "inverse object detection" approach work in principle?
2. What are the limitations?
3. How can the training data mismatch be overcome through data augmentation and other training techniques?

What this chapter will *not* do is investigate the small details of network architecture and optimisation aspects of training (learning rates, etc.). One can spend hours fine-tuning these hyperparameters to eke out a few percentage points of performance for a specific dataset, but this is not the aim of this study. It is assumed that most published networks have already gone through this process and it is likely that little value will be gained from fiddling with parameters when training and testing on very different domains anyway.

The proposed object detection concept is shown in Fig. 5.1. The system takes 3-channel RGB images as input and processes them through a semantic segmentation

network to generate a class probability distribution for each pixel. The probability distributions are then processed to create a segmentation of the scene based on contextual knowledge of maritime scenes. Object detections are then extracted from this scene model.

5.2 Semantic segmentation networks

5.2.1 Selected networks

Semantic segmentation networks were selected from recent literature which have been designed to be efficient, motivated by use in real-time applications such as autonomous driving. The criteria for selection were:

- Good performance on benchmark datasets
- Fast inference speed (for real-time applications)

The networks are fully convolutional and do not rely on post-processing of the network output (e.g. using CRF refinement [79]) to obtain high accuracy. Both of these features are important for reducing the number of network parameters and keeping inference speed fast. Being fully convolutional also means they can be applied to input images of any size, irrespective of the size of the training images. This is useful for real-world applications, where the input data may not be the same resolution as the training data.

Seven networks were selected for the initial round of evaluation (see Table 5.1). The networks obtain a range of accuracy performances on benchmark datasets such as CamVid [41] and CityScapes [52]. No results for UNet could be found for the Cityscapes dataset but it is included because it has performed well in the medical imaging domain [216].

5.2.2 Baseline performance on CamVid dataset

The results from the literature (Table 5.1) indicate that these networks are interesting for this study. However, it is difficult to make meaningful comparisons between networks using values which are reported in different sources, as the implementation details are not fully known and runtime speeds are affected by computing platforms and frameworks. To further refine the selection, the networks were implemented in the same framework and evaluated on the same computing platform on a common dataset.

Table 5.1 Semantic segmentation networks shortlist. Number of parameters and FPS for the CityScapes [52] dataset (*values from implementation in this thesis, †values reported in [139], ‡values reported in [183], **values reported in [245])

Network	Year	Params*	FPS		mIoU
			1024 × 512	2048 × 1024	
UNet [184]	2015	31.0M	-	-	-
SegNet [12, 13]	2015	24.9M	1.6‡	-	57.0‡
ENet [159]	2016	0.37M	76.9†	20.4‡	58.3‡
ERFNet [182, 183]	2017	2.07M	41.7†	11.2‡	69.7‡
ESPNet [147]	2018	0.35M	112.9†	-	60.3†
EDANet [139]	2018	0.67M	81.3†	-	67.3†
ICNet [245]	2018	6.70M	-	30.3**	70.6**

Training data

To get results which would indicate suitability for the task of learning from limited data, the CamVid dataset [41] was selected to benchmark the performance of the networks under the conditions of limited training data, but high correlation between training and test data. The training, validation and test splits were taken from the authors' online repository[1]. The number of train, validation and test images are 367, 101 and 233, respectively, which corresponds to approximately 53%, 14% and 33% (a 60% / 10% / 30% split is commonly used). The dataset has 11 classes (plus a 12th 'void' class, which is ignored in calculation of the loss) and a resolution of 480 × 360. This dataset is particularly suitable for preliminary investigations as it is of similar size to the amount of maritime training data that can be obtained from other sources (see Section 5.3). Fig. 5.2 shows some example images and their segmentation labels from the CamVid dataset.

Implementation and training

For fair comparison of performance (particularly speed), all the models were implemented in the PyTorch framework based on the author's published code, using the original source code where possible. The network implementations were taken from the following sources:

[1] https://github.com/alexgkendall/SegNet-Tutorial/tree/master/CamVid

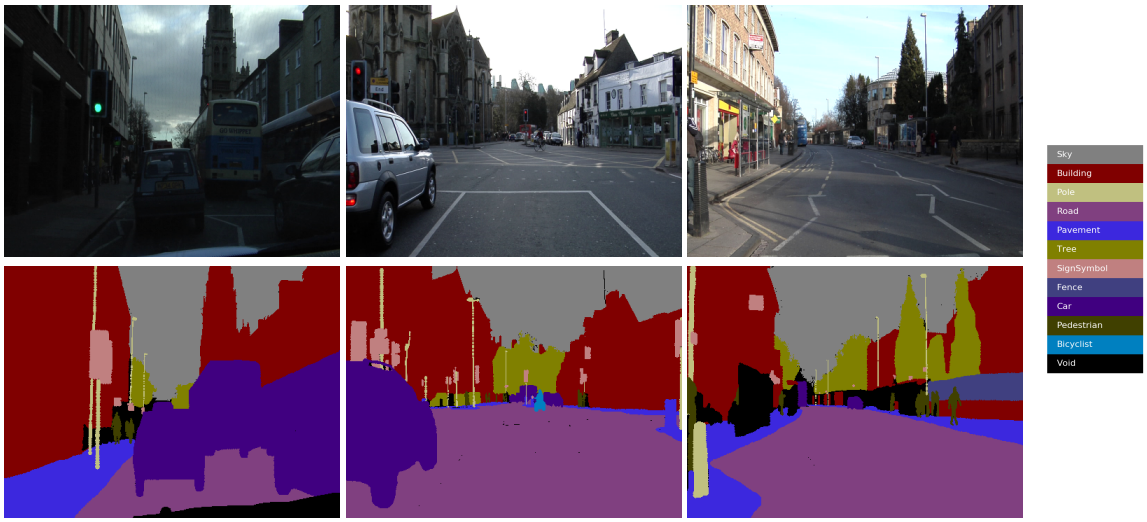


Fig. 5.2 Example CamVid [41] images and groundtruth labels

UNet[2] SegNet[3], ENet[4], ERFNet[5], ESPNet[6], EDANet[7]. The ICNet source code is not publicly available so this was implemented from scratch following the paper. The hyperparameters, class balancing scheme and training protocol used for training are the ones described in each of the original papers (see Table 5.2).

Training was performed on an Alienware laptop with an 8-core 2.6GHz Intel® Core™ i7 CPU and 16GB RAM, with an externally connected NVIDIA® GeForce® GTX™ Titan X GPU with 12GB memory. The networks were all trained end-to-end with a multi-class cross entropy loss using a batch size of 8, apart from SegNet and UNet, which were trained with batch sizes of 4 and 2, respectively, due to GPU memory limitations. The images were presented to the networks at 3 different sizes: 480 × 360 (original size), 352 × 288 (0.75×) and 608 × 448 (1.25×). Random horizontal flips were used as basic data augmentation. Class balancing was used to mitigate the effects of unbalanced datasets on training. This is especially important in semantic segmentation where there might be hundreds of millions of pixels of some classes and only thousands of pixels of others. The networks use two methods for class balancing: median frequency and a custom method first proposed in the ENet paper [159]:

[2]https://lmb.informatik.uni-freiburg.de/people/ronneber/u-net
[3]https://github.com/alexgkendall/SegNet-Tutorial
[4]https://github.com/TimoSaemann/ENet
[5]https://github.com/Eromera/erfnet
[6]https://github.com/sacmehta/ESPNet
[7]https://github.com/shaoyuanlo/EDANet

Table 5.2 Semantic segmentation network training setup

Network	Class Balancing	Optimisation
UNet [184]	Median Frequency	SGD; momentum: 0.99; lr: 0.0001; step, γ: 0.1; L2 reg.: *none*
SegNet [13]	Median Frequency	SGD; momentum: 0.9; lr: 0.001; L2 reg.: 0.0005
ENet [159]	ENet custom $c = 1.02$	ADAM; βs: 0.9, 0.999; lr: 0.005, step γ: 0.1; L2 reg.: 0.0002
ESPNet [147]	ENet custom $c = 1.10$	ADAM; βs: 0.9, 0.999; lr: 0.0005, step, γ: 0.5; L2 reg.: 0.0005
ERFNet [183]	ENet custom $c = 1.10$	ADAM; βs: 0.9, 0.999; lr: 0.0005, poly, γ: 0.9; L2 reg.: 0.0001
EDANet [139]	ENet custom $c = 1.12$	ADAM; βs: 0.9, 0.999; lr: 0.0005, poly, γ: 0.9; L2 reg.: 0.0001
ICNet [245]	ENet custom $c = 1.10$	ADAM; βs: 0.9, 0.999; lr: 0.01, poly, γ: 0.9; L2 reg.: 0.0001

Median frequency

$$w_i = \frac{med.\ freq.}{f_i} \tag{5.1}$$

where f_i is the frequency of class i, *med. freq.* is the median of all class frequencies, and w_i is the weight applied to class i.

ENet custom

$$w_i = \frac{1}{\ln(c + p_i)} \tag{5.2}$$

where p_i is the probability of class i, c is a hyperparameter which restricts the range of class weights, and w_i is the weight applied to class i.

Evaluation metrics

For quantitative evaluation of the semantic segmentation output, accuracy and Intersection over Union (IoU) are used, as defined in [140]. Accuracy is the proportion of pixels in the image which were correctly classified and is computed per class and globally. IoU is the ratio of overlap between the predicted and groundtruth segmentation and their total area. This is also computed per class and an average over classes is taken to give the mean

IoU (mIoU). In the following, n_{ij} is the number of pixels of class i predicted to belong to class j, there are n_{cl} different classes, and $t_i = \sum_j n_{ij}$ is the total number of pixels of class i.

$$\text{Per-class accuracy} = \frac{n_{ii}}{t_i} \tag{5.3}$$

$$\text{Global accuracy} = \frac{\sum_i n_{ii}}{\sum_i t_i} \tag{5.4}$$

$$\text{Per-class IoU} = \frac{n_{ii}}{t_i + \sum_j n_{ji} - n_{ii}} \tag{5.5}$$

$$\text{Mean IoU} = \left(\frac{1}{n_{cl}}\right) \sum_i \frac{n_{ii}}{t_i + \sum_j n_{ji} - n_{ii}} \tag{5.6}$$

Results and analysis

Fig. 5.3 shows the loss and accuracy for training and validation. All the networks converge relatively quickly (within ~150 epochs). UNet converges more slowly, and to a higher loss and lower accuracy. ICNet converges at a similar rate to the others, but also does not reach the same loss or accuracy. The convergence of UNet can be explained by the fact that it has a very large number of parameters. SegNet is also a large architecture, but the max unpooling connections are effective in routing the loss signal to the lower layers so that they train faster.

Table 5.3 compares the results achieved in this experiment against those reported in the literature. In this test, no effort was made to optimise the training regime so the results are lower than those reported, as expected. An interesting observation is that the more recent networks typically report better mIoU scores, but in this test that was not the case. For example, the most recent network, ICNet, ranked 6th out of 7, falling behind one of the oldest networks, SegNet. Possible reasons for this are discussed at the end of this section.

Fig. 5.4 compares the different properties and performance of the networks. As expected, the networks with fewer parameters are also the fastest to train and run (Fig. 5.4e and 5.4f). ENet and ERFNet are slightly slower compared to others of similar size (EDANet and ICNet) resulting from differences in architectural structures. For example, dilated convolutions can have the same number of parameters as standard convolutions, but fewer strides are needed to cover the same input volume. Network operations can be

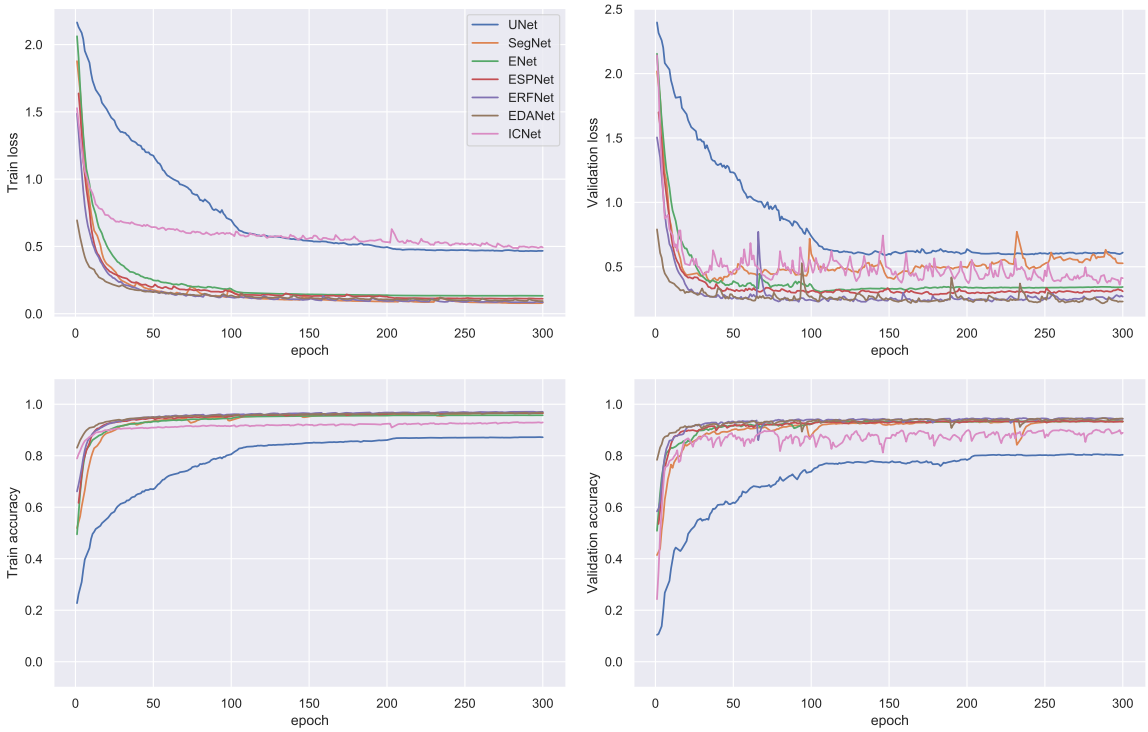


Fig. 5.3 CamVid baseline training and validation curves

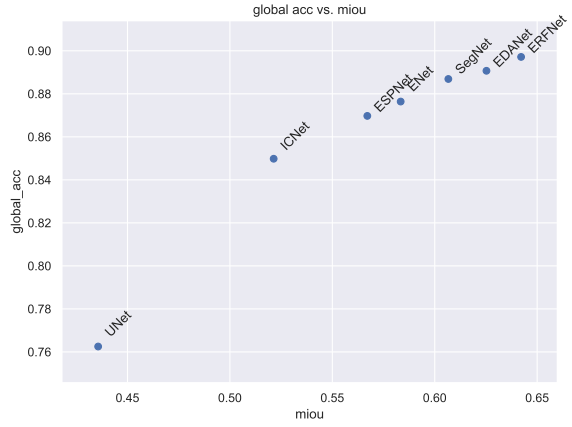
Table 5.3 Reported vs. achieved results for CamVid baseline test

Network	Reported mIoU	Achieved mIoU	Rank	Achieved FPS 480 × 360	No. params.
UNet [184]	-	40.4	7	25.5	31.0M
SegNet [12, 13]	55.6	45.8	5	37	24.9M
ENet [159]	51.3	48.8	4	73	0.37M
ESPNet [147]	55.64	51.8	2	112	0.35M
ERFNet [182, 183]	-	52.0	1	83	2.07M
EDANet [139]	62.6	49.3	3	116	0.67M
ICNet [245]	67.1	44.6	6	95.2	6.70M

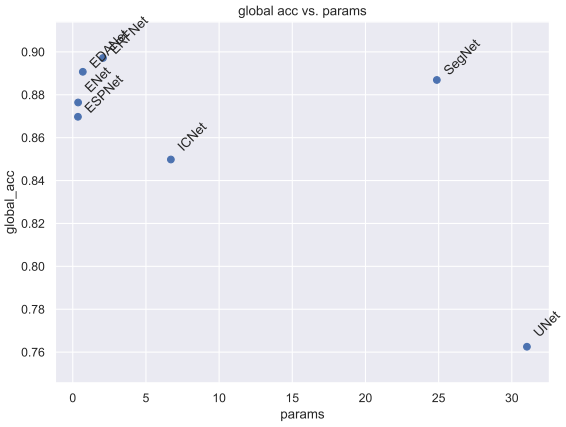
optimised differently by the software library and GPU which could also lead to differences in speed. ERFNet achieves higher mIoU and accuracy than the other networks so although it is slightly larger and slower, it is using its parameters more effectively. Generally, the smaller networks achieve the best accuracy and mIoU scores. This is due to the recent developments in architectural structures which make more efficient use of parameters. When comparing speed with mIoU (Fig. 5.4c), there is more variability in the performance of the networks so selecting the best network for a real-time application is not straightforward.

The segmentation performance is broken down by class in Fig. 5.5. UNet is the lowest scoring across the classes, but its accuracy score can be higher (e.g. for Pole and Sign in Fig. 5.5a). The accuracy metric has a bias towards the negative case (i.e. how often the class is not present) which can give misleading results for rarer classes. The mIoU scores (Fig. 5.5b) do not have this bias and here it is clear that UNet does particularly badly with the rarer classes (Pole, Sign, Fence, Pedestrian and Bicyclist), hence it has a low mIoU. The results for the other networks do not vary significantly from class to class and are consistent with the overall accuracy and mIoU performance discussed above.

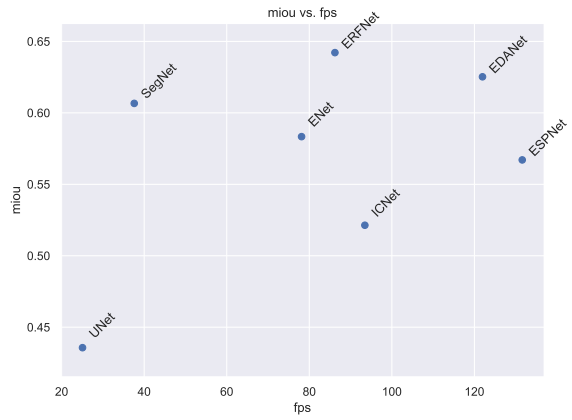
To analyse the runtime inference speed in more detail, the networks were also run on larger images, as the CamVid data is lower resolution (480 × 360) than the target IPATCH data (1920 × 1080). Fig. 5.6 shows speed in terms of processing time per frame and FPS for different image sizes. The general trend is as expected (i.e. that larger images take longer to process) but it is interesting that networks decrease at different rates and sometimes switch order (this is more noticeable with FPS in the right-hand side of Fig. 5.6). For example, EDANet is faster than ICNet with the 480 × 360 images but ICNet is faster with the larger images. This is likely due to how the software library and GPU are able to optimise different network operations with different amounts of data. The main conclusion from Fig. 5.6 is that real-time performance on the IPATCH data will not be achievable with all the networks in the shortlist.



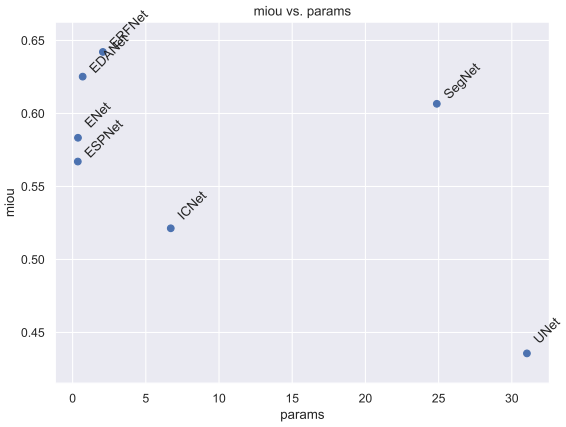
(a) Accuracy vs. mIoU



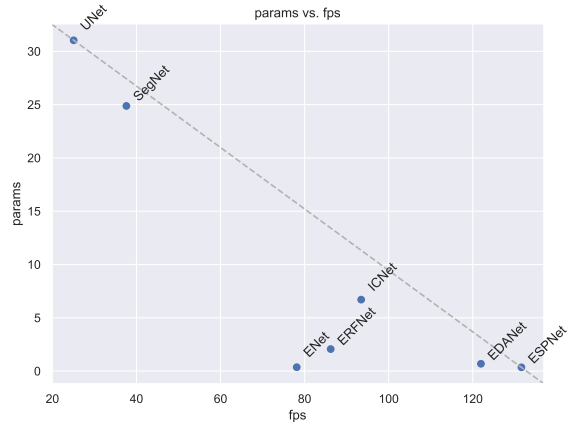
(b) Accuracy vs. number of parameters



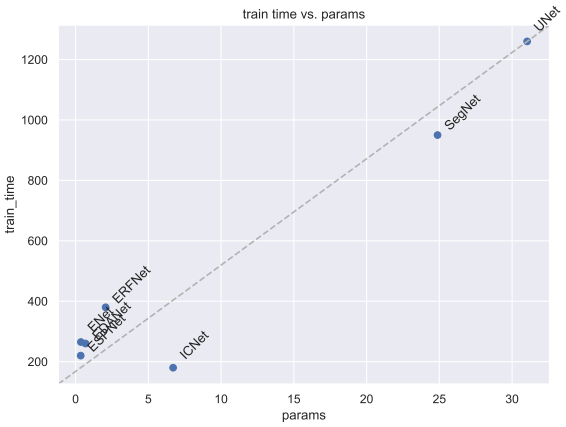
(c) mIoU vs. FPS



(d) mIoU vs. number of parameters

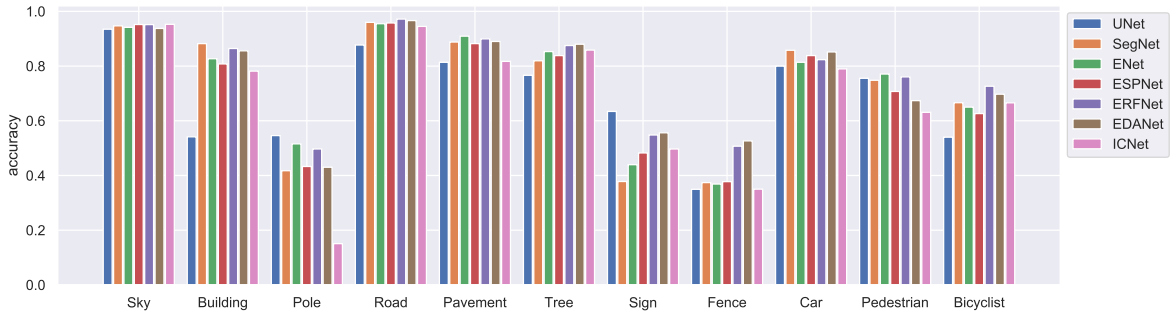


(e) Number of parameters vs. FPS

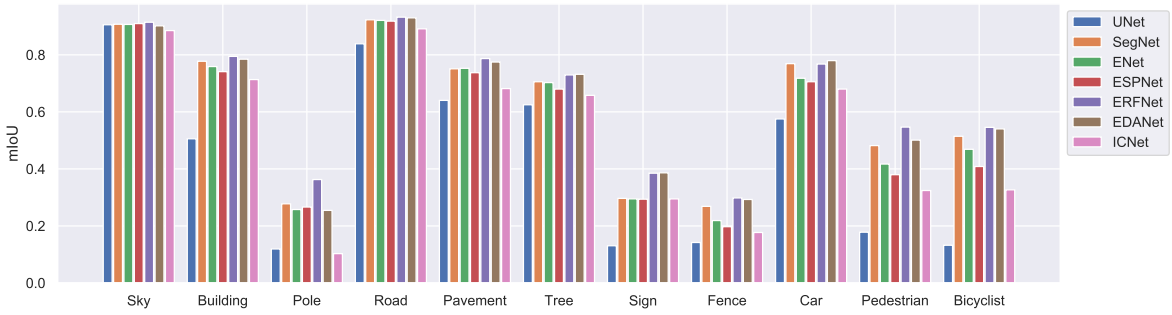


(f) Training time vs. number of parameters

Fig. 5.4 Semantic segmentation network properties based on CamVid baseline experiments



(a) Per-class validation accuracies



(b) Per-class validation IoUs

Fig. 5.5 CamVid baseline validation per-class accuracies and IoUs

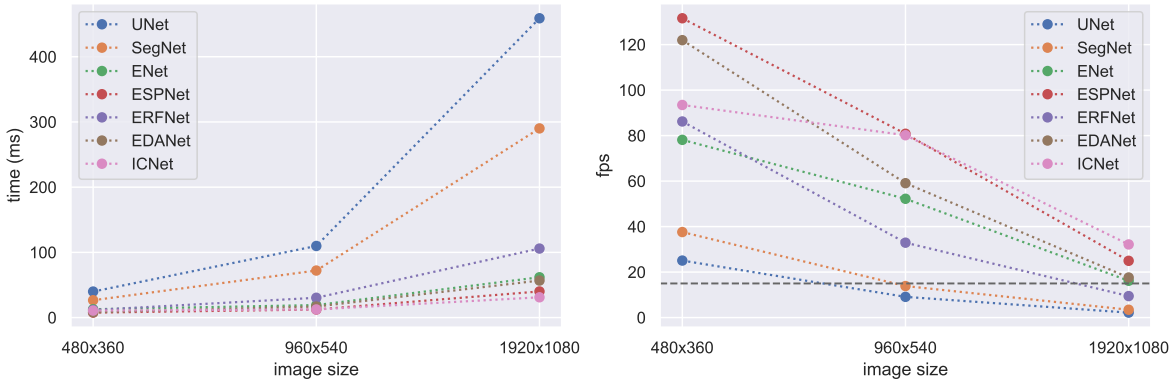
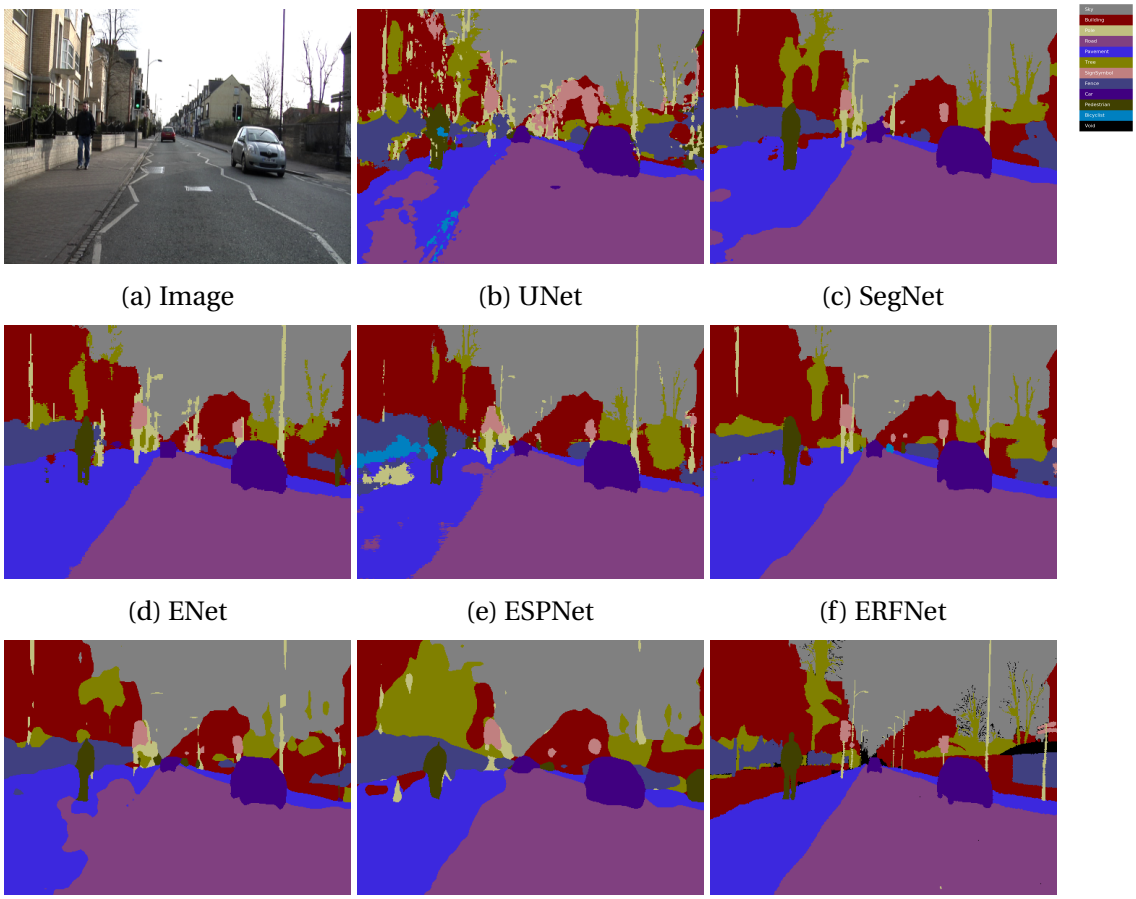


Fig. 5.6 Inference speeds for the networks on different image sizes. The horizontal line in the right-hand plot marks 15 FPS, which could be considered a target minimum speed for real-time applications.



(a) Image (b) UNet (c) SegNet

(d) ENet (e) ESPNet (f) ERFNet

(g) EDANet (h) ICNet (i) Groundtruth

Fig. 5.7 Qualitative results for the networks on CamVid in the baseline experiment.

Downselection of networks

Based on the CamVid baseline results above, it is clear that UNet and SegNet do not really compete with the more recent and efficient networks. The output of UNet in particular is visually not as good as the other networks (Fig. 5.7). As they are bigger architectures, they take longer to train and run but do not give good enough segmentation performance to compensate.

The performance achieved by ICNet is unexpected. Its segmentation output is visually not well-defined (see Fig. 5.7h), and the scores achieved are much lower than those reported in [245]. The scores are also low compared to some of the less sophisticated and older networks, which is surprising.

ICNet is one of the networks that can make use of a pre-trained encoder from other work (PSPNet [246]). It's possible that, without pre-training on e.g. ImageNet, it is not able to learn good enough features on a small training set when trained end to end. However, other networks did not have this problem in this baseline test.

Another reason could be due to the weights used in the multi-scale cross-entropy loss. The image is input to the network in low, medium and high resolution branches. Deep supervision is implemented by computing cross-entropy losses for each branch which are combined into an overall loss using different weights. The weight values were determined empirically, so they may not generalise well to new datasets or may not be suitable for end to end training.

Further investigation of pre-training and the multi-scale loss weights in ICNet is left for future work. Based on the preliminary tests in this section, UNet, SegNet and ICNet are not included in further experiments.

5.3 Training on data from the ADE20k dataset

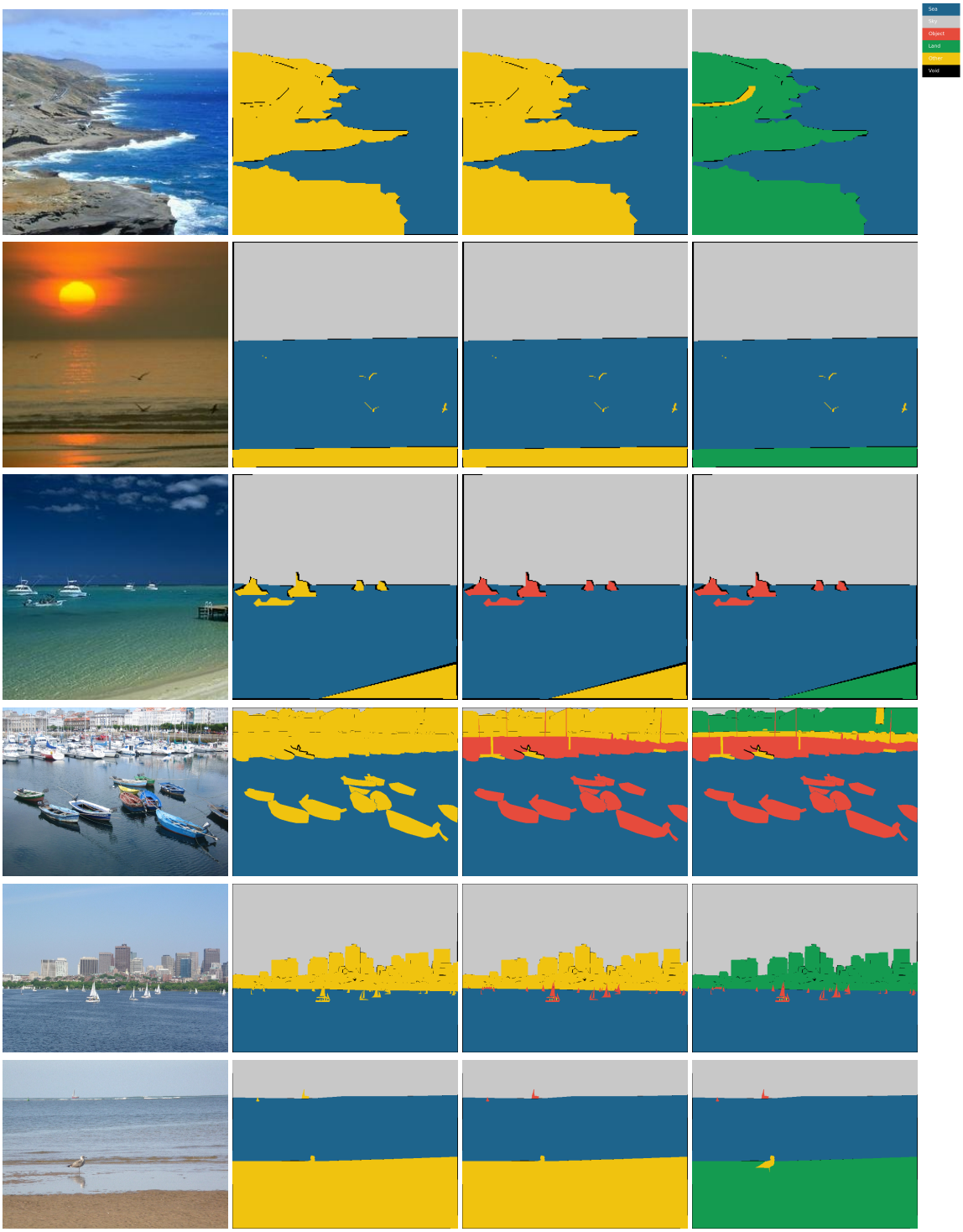
5.3.1 Training data

The ADE20k dataset [248] was selected because it is the largest available scene (rather than object) semantic segmentation dataset that contains the classes of interest (sea, boat, etc.). It is not ideally suited to the maritime surveillance task, but it is the only dataset currently available which covers the relevant classes with sufficient pixel-level groundtruth for training semantic segmentation networks. However, it also contains many other classes which are not relevant to maritime surveillance.

A subset of the data was therefore created for training by manually extracting images which primarily contain classes from maritime surveillance (i.e. sea, sky, boats, buoys, etc.). Some images were further excluded because they are not suitable: for example, an image that contains a painting of a boat on a wall in a room rather than a real boat scene, or images where sky is only visible through a window. This process resulted in 434 images with median dimensions of 300×256 pixels (note that this is significantly smaller than the target domain of 1920×1080 images). All the classes in the 434 images are then mapped to a new, smaller set of classes which are relevant for the task at hand (correcting the labels manually, where necessary).

Initially, all classes were mapped to one of *Sea*, *Sky*, and *Other* (maintaining the 'void' class wherever present). Two other mappings were also investigated – one which distinguishes *Object* separately from *Other*, and one with *Object*, *Land* and *Other* as distinct classes – to see if this makes it easier or harder for the network to distinguish objects of interest from the rest of the scene. Note that the *Object* class refers to maritime objects, i.e. those that are found on the surface of the sea and make up the targets for detection in maritime surveillance. This includes large ships, speedboats, sailing boats, buoys, and so on. Any other objects (e.g. birds, vehicles on land) are mapped to the *Other* class. Examples of images in the subset for training can be seen in Fig. 5.8.

The three versions of the subset were split into train, validation and test sets for the experiments using a standard split of approximately 60% / 10% / 30%. The split process applied random stratified sampling based on image size and scene category to minimise the difference in distributions between the splits. Scene category is provided in the ADE20k annotation data and image size was approximated by using K-Means clustering to group



(a) Images (b) 3-plus-1 labels (c) 4-plus-1 labels (d) 5-plus-1 labels

Fig. 5.8 Example training images and class mappings from the 434-image maritime subset of ADE20k [248].

the images into 'small', 'medium' and 'large'. The properties of the data for the 3 class mappings can be found in Tables 5.4a to 5.4c.

Table 5.4 Dataset properties for the 3 class mappings of the maritime subset of ADE20k

(a) 3-plus-1 class mapping

	Sea	*Sky*	*Other*	*Void*
No. Image Occurrences	430	403	423	433
Total proportion (all images)	32.6%	30.1%	35.7%	1.6%
Mean proportion (per image)	34.5%	33.2%	30.4%	1.9%

(b) 4-plus-1 class mapping

	Sea	*Sky*	*Object*	*Other*	*Void*
No. Image Occurrences	430	403	180	421	433
Total proportion (all images)	32.5%	30.1%	6.2%	29.6%	1.6%
Mean proportion (per image)	34.5%	33.2%	3.9%	26.5%	1.9%

(c) 5-plus-1 class mapping

	Sea	*Sky*	*Object*	*Land*	*Other*	*Void*
No. Image Occurrences	430	403	180	392	233	433
Total proportion (all images)	32.5%	30.1%	6.2%	19.8%	9.8%	1.6%
Mean proportion (per image)	34.5%	33.2%	3.9%	19.6%	6.9%	1.9%

5.3.2 Implementation and training

The implementations and hyperparameters for the networks are the same as those described in Sec. 5.2.2. ESPNet was modified to allow fewer than K output classes, where K is a hyperparameter of the network set to 5. K controls the number of parallel branches in the ESP block. The final ESP block in the network projects its input volume to the number of output classes, C. Each of the K parallel branches must produce an output of at least 1 channel. When these are concatenated, the number of output channels is 5, hence will be greater than C when there are fewer than 5 classes. The ESP module was therefore modified to automatically adjust to use fewer parallel branches if the number of output channels was less than K.

All networks were trained end-to-end with batches of 8 images, scaled to 4 different sizes to capture information at different scales: 672 × 512, 512 × 384, 416 × 320 and 320 × 256 (keeping aspect ratio by randomly cropping, if necessary). These sizes are based on the fact that the image is downsampled up to 5 times in some networks, so values were selected that are multiples of 2^5 with approximately the same aspect ratio. For the baseline, only random horizontal flips are applied as data augmentations. Training was performed on the same computing platform as before (Sec. 5.2.2).

K-fold cross-validation

Ultimately, the aim is to use all available data for training, especially as the amount of training data is limited. However, the hyperparameters and training regime need to be checked to make sure the network is learning well on the data (i.e. loss is decreasing) and will not overfit. The latter is particularly important for small datasets. A k-fold cross-validation approach was adopted to address this. This is a way to ensure that the initial selection of the train, validation and test splits was not, by chance, particularly good or bad for training. It gives confidence that the splits are, in fact, representative, and that conclusions on one split will also be valid on other splits. The process also tests if the hyperparameters are appropriate and reveals how long the network takes to converge. When training with the full data, where there is no validation set to test for over- or under-fitting, one can be confident that the network has converged at around the same point.

Each network was trained on k = 5 different splits of the ADE20k subset. The accuracy and mIoU curves from training are shown in Fig. 5.9. The mean scores over the 5 different splits are plotted with error bars showing the standard deviation. All the networks reached convergence by 500 epochs, and there is small variation between the splits. Some variation is expected, due to the small number of images. This exercise validates the hyperparameters for each network and means that overfitting will not occur when training for 500 epochs on the full ADE20k subset (434 images).

Maritime surveillance test data ('MarSemSeg')

In this work, two key aspects of the approach are a) to use as much training data as possible and b) test how well the trained models will generalise to maritime surveillance data. The first point means there is no data left over to evaluate on and the second

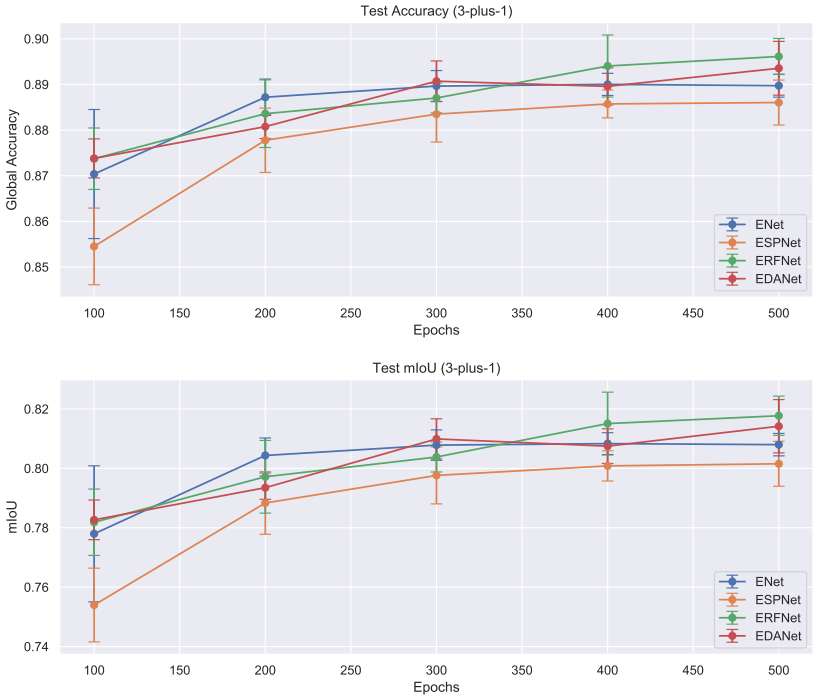
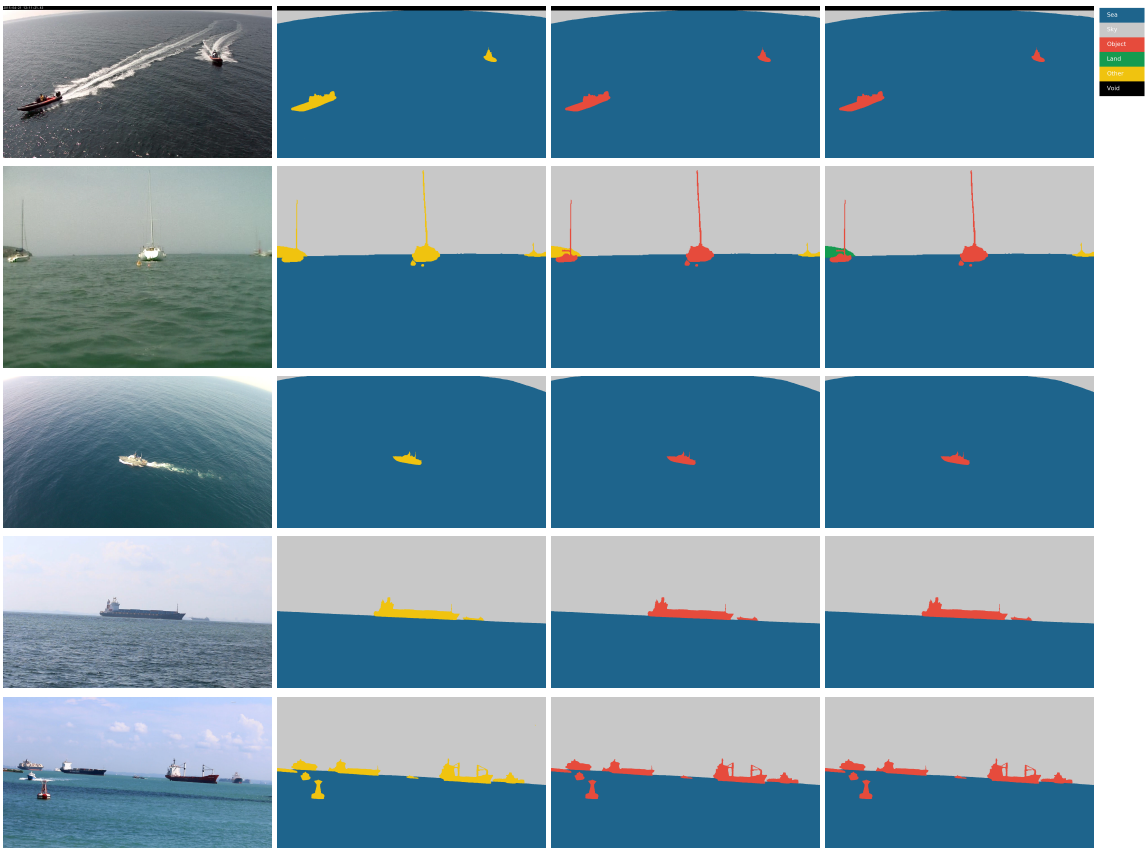


Fig. 5.9 k-fold cross-validation training for 3-plus-1 mapping

point arises because the ADE20k images are not very representative of the kind of scenes in the maritime surveillance domain (even though they contain the right classes, they are qualitatively different). For this reason, a small test set was created by producing semantic segmentation groundtruth for images taken from the maritime datasets. The sequences from Chapter 4 were used, as saliency groundtruth had already been created, and this could be used to 'bootstrap' the creation of the semantic segmentation labels. the maritime semantic segmentation test set – or 'MarSemSeg' for short – consists of 70 images sampled from the 7 sequences in Chapter 4 (10 images sampled uniformly through each sequence to minimise similarity of images and capture a range of object sizes and scene features). In the rest of this chapter, the networks were trained using the full set of 434 images. The k-fold cross-validation process gives confidence that the networks would not overfit within 500 epochs. The MarSemSeg test is then used to measure performance and the ability to generalise to the maritime surveillance domain. Fig. 5.10 shows some examples from the MarSemSeg dataset and Table 5.5 lists the properties of the 3 class mappings.



(a) Images (b) 3-plus-1 labels (c) 4-plus-1 labels (d) 5-plus-1 labels

Fig. 5.10 Examples of MarSemSeg images and groundtruth labels

Table 5.5 Dataset properties for the 3 class mappings of MarSemSeg

(a) 3-plus-1 class mapping

	Sea	*Sky*	*Other*	*Void*
No. Image Occurrences	70	68	65	30
Total proportion (all images)	71.2%	26.4%	1.4%	1.0%
Mean proportion (per image)	70.22%	27.0%	1.3%	1.5%

(b) 4-plus-1 class mapping

	Sea	*Sky*	*Object*	*Other*	*Void*
No. Image Occurrences	70	68	65	12	30
Total proportion (all images)	71.2%	26.4%	1.4%	0.03%	0.97%
Mean proportion (per image)	70.2%	27.0%	1.15%	0.11%	1.55%

(c) 5-plus-1 class mapping

	Sea	*Sky*	*Object*	*Land*	*Other*	*Void*
No. Image Occurrences	70	68	65	2	12	30
Total proportion (all images)	71.2%	26.4%	1.38%	0.0%	0.02%	0.97%
Mean proportion (per image)	70.2%	27.0%	1.17%	0.01%	0.1%	1.55%

5.4 Experiments

In this section, a number of factors are investigated to see which approaches will improve the ability of a semantic segmentation networks to learn from non-ideal data (434 images selected from the ADE20k scene parsing dataset) and be applied to maritime surveillance data (MarSemSeg). The evaluation metrics for semantic segmentation are the same as those described in Section 5.2.2.

5.4.1 Number of classes

The three sub-set mappings (3, 4 and 5 classes) were used to investigate the performance of each of the remaining networks, and to see if training on more or fewer classes leads to learning better features. Intuitively, learning fewer classes should be easier, but forcing the networks to distinguish between more classes might lead to more discriminative features which generalise better to new data.

Looking at the results for the 4 and 5 class sub-sets (Fig. 5.11), it can be seen that IoU for the classes which are not sea or sky drops dramatically. All networks struggle to distinguish between Objects, Land and Other because of the small amount of training data. Looking at the confusion matrices (Fig. 5.13), the Object and Other classes are most confused, suggesting that there is insufficient inter-class variance and/or too much intra-class variance for the networks to learn discriminating features. The segmentation examples in Fig. 5.14 show how all of the networks tend to detect objects as mixtures of multiple classes (column (b) and (c)) which would make object detection more difficult. Based on these results, the decision was made to proceed with the 3 class subset for the rest of the experiments, where Objects, Land and Other are aggregated in the Other class.

In terms of individual network performance, ERFNet performs best overall, with EDANet sometimes performing better and often being second best. The qualitative results in Fig. 5.14 support the quantitative analysis. On the 3-class subset, EDANet does better than ENet and ESPNet in overall mIoU, and gets higher IoU on the Other class. It is slightly surprising that ESPNet achieved the lowest scores overall. It's possible that the modifications to the ESP module to handle fewer than 5 classes have adversely impacted ESPNet's performance. On the basis of these results, EDANet is selected for further experiments. Whilst it does not achieve quite as high mIoU score as ERFNet, it



(a) 3-plus-1 (b) 4-plus-1 (c) 5-plus-1

Fig. 5.11 Results of the 4 networks on MarSemSeg test set, trained on full ADE20k maritime subset

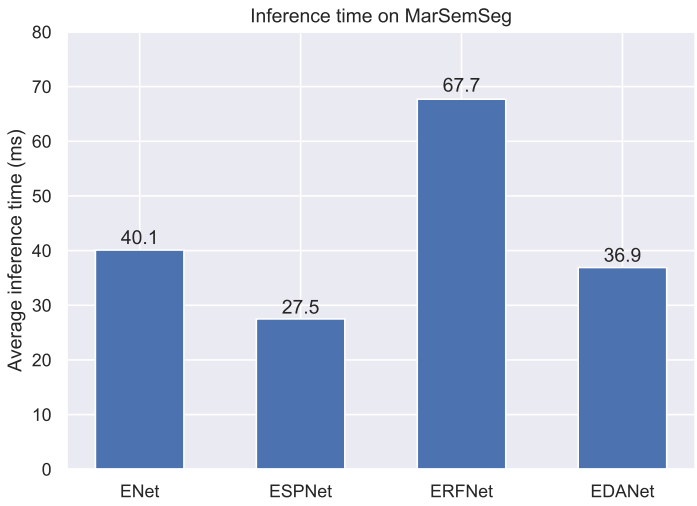
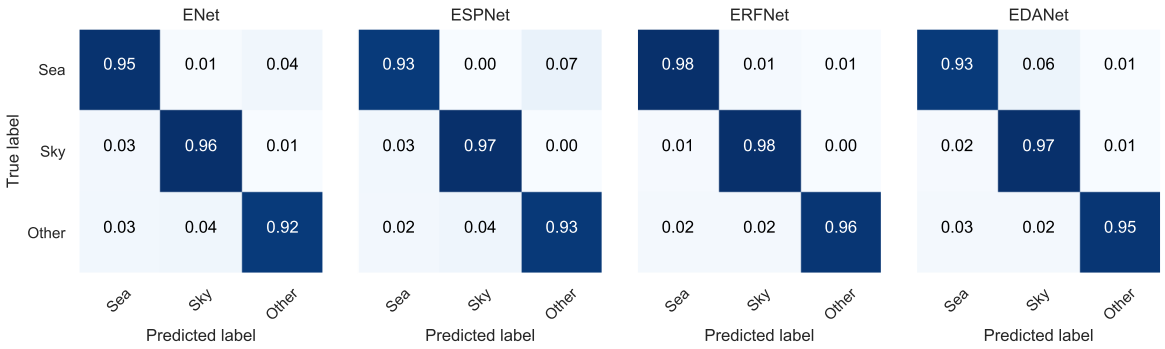
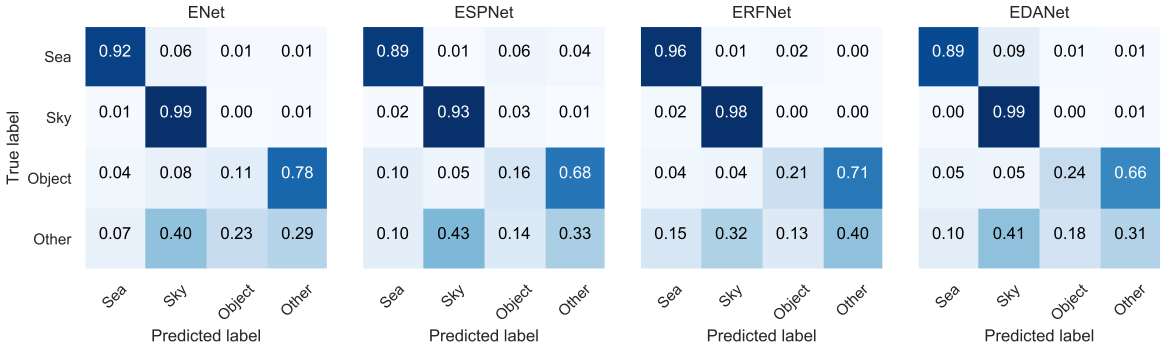


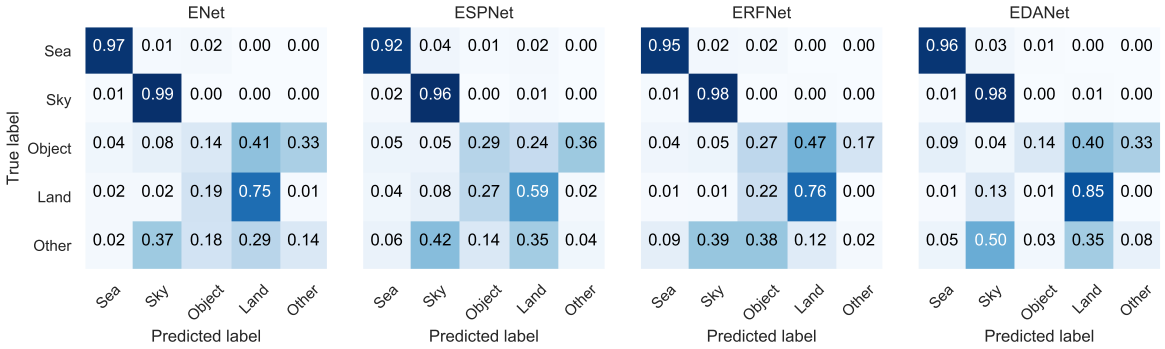
Fig. 5.12 Inference time on MarSemSeg



(a) 3-plus-1

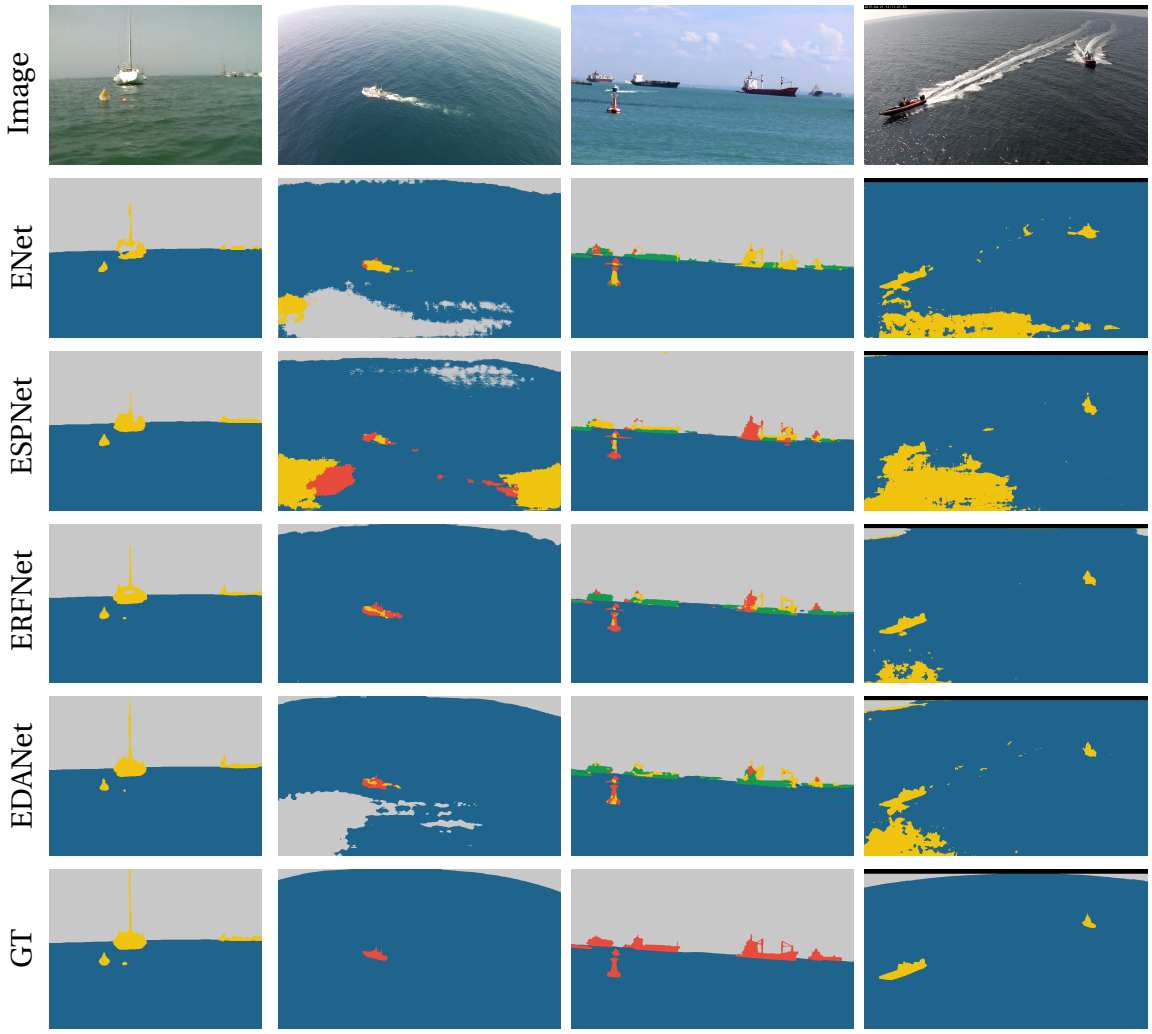


(b) 4-plus-1



(c) 5-plus-1

Fig. 5.13 Confusion matrices for the 4 networks on MarSemSeg test set, trained on full ADE20k maritime subset



(a) 3-plus-1 (b) 4-plus-1 (c) 5-plus-1 (d) 3-plus-1

Fig. 5.14 Example segmentation output for the networks on MarSemSeg

has a faster inference time (see Fig. 5.12), which makes it more attractive to real-time applications.

5.4.2 Data augmentation

Data augmentation is the process of applying transformations to a dataset to create modified versions of the original images, thereby expanding the total number of different images that the network sees during training. It is especially important when working with small datasets, such as the ADE20k subset used in this study.

Data augmentation tries to bridge the gap between training data and test data by using expert knowledge to make the training distribution closer to the test distribution. It teaches a network about invariances in the data domain so that the learned model is also invariant to these features, thereby improving its performance on unseen data. One thing that it cannot do, though, is provide more examples of objects of different types or from different viewpoints. For example, if there is no image of boat taken from the stern in the dataset, this cannot be simulated through data augmentation.

In the baseline experiments above, only very basic augmentation transforms were used: random horizontal flips and resizing to four different predetermined input sizes. These can be considered "standard" augmentations that are done with even very large datasets [82, 90, 123]. This section investigates the effectiveness and importance of more extensive data augmentation strategies in the task of adapting networks trained on general scene parsing data to use on maritime surveillance data. The data augmentations are grouped into three categories:

1. **Photometric:** e.g. saturation, brightness, contrast, colour balance – these are designed to simulate the effects of different camera sensors and lighting conditions
2. **Geometric:** e.g. rotation, shear, perspective transforms, warp/distortion – these are designed to simulate different camera optics and viewpoints
3. **Noise:** e.g. blur, additive Gaussian noise, pixel dropout, compression – these are designed to simulate imaging artefacts of real-world systems

To teach the network about invariances which are relevant for the maritime surveillance domain, seven augmentations were implemented (see Appendix A for full details), in addition to horizontal flips and multiple input sizes:

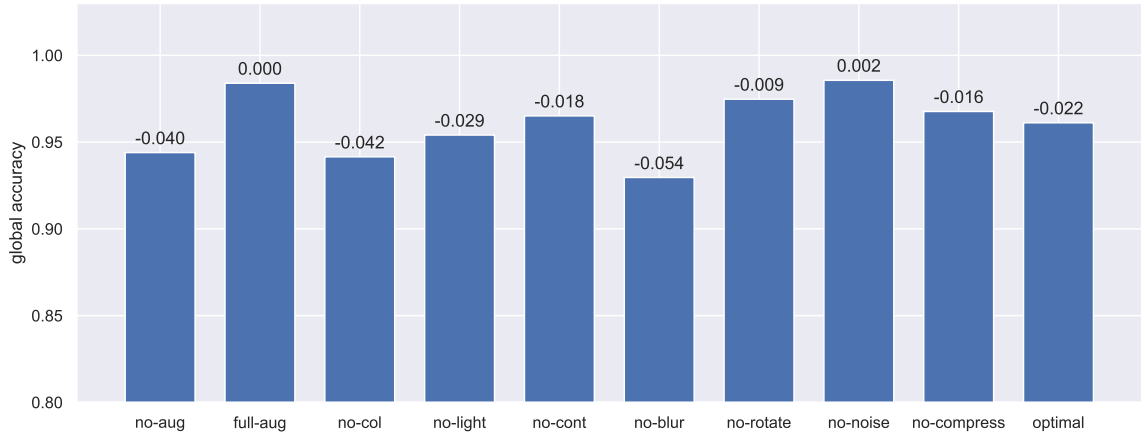
- Brightness (1), contrast (2) and colour (3) variations can simulate the changes in appearance under different lighting conditions, in particular when there is bright sun glare present, or when lighting is low (cloudy day or evening sequences).
- Rotation (4) of images around the centre can simulate the change in viewpoint due to the motion (roll and pitch) of vessels at sea.
- Blur (5) is common in real systems if the camera is not properly focused.
- Image noise (6) can appear in cameras under real-world conditions (e.g. due to electromagnetic interference, video decoding errors, or low-light conditions).
- Compression artefacts (7) are common due to the need to transmit high-resolution video over limited bandwidth.

An ablation study was conducted to assess the importance of each of these augmentations. First, the network was trained with the baseline augmentations only and then with all 7 augmentations applied. The network was then run 7 more times, each time removing one of the augmentations to see how performance was affected. The results of this process are show in Fig. 5.15.

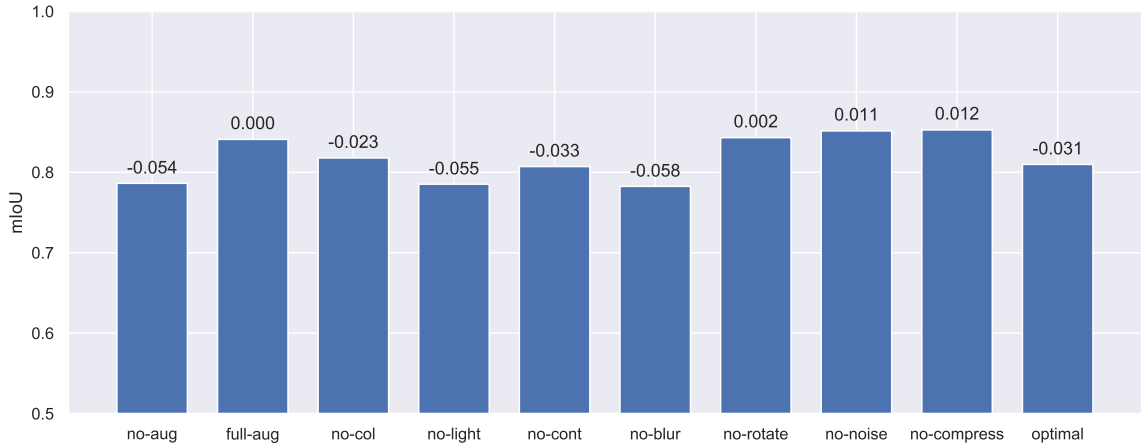
Applying all the augmentations together (*full-aug*) showed an increase in performance. Removing some augmentations (e.g. colour and blur) decreased performance compared to the full set, indicating that they are providing value. In comparison, the removal of some augmentations (e.g. rotation, noise and compression artefacts) actually *increased* performance, indicating that they might be hindering generalisation, rather than improving it.

An 'optimal' set was created where these augmentations (rotation, noise and compression artefacts) were removed. Although performance was better compared to the baseline, it did not reach as high as when all the augmentations were applied. The conclusion is that the interaction of augmentations is not trivial or additive. Some augmentations may increase the accuracy on some classes, but decrease on others. This can be seen in the per-class IoU scores in Fig. 5.15c. Augmentations may also interact with each other in a non-predictable way on different classes.

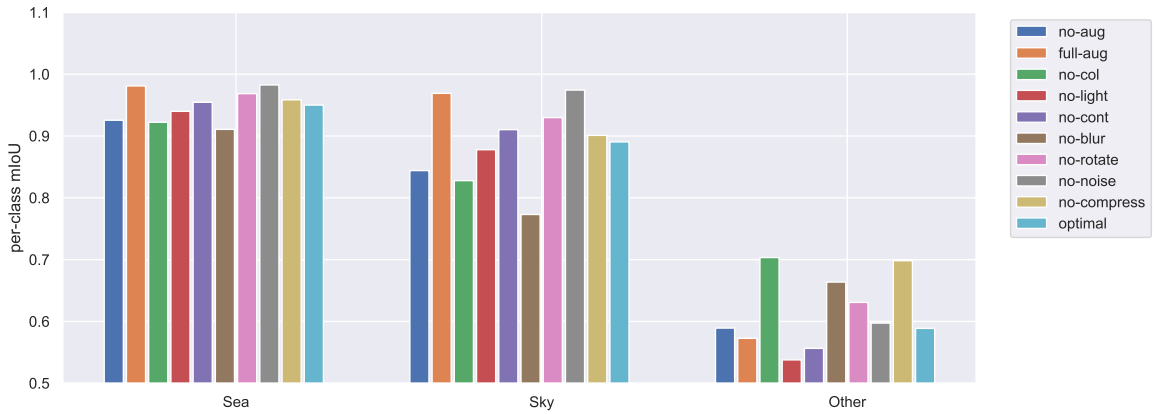
It is clear that augmentation is beneficial, but it cannot categorically be said that some augmentations are better or 'best' – there is no optimal set. It is highly dependent on both the training data and eventual application domain (test) data. In the next section, a different approach is investigated, whereby the network is trained on a different but related task which encourages it to use spatial information in the scene.



(a) Accuracy



(b) mIoU



(c) Per-class IoU

Fig. 5.15 Ablation study results showing (a) global accuracy, (b) mean IoU over classes and (c) per-class IoU scores. In (a) and (b), the numbers above each bar show the relative change compared to the full set of augmentations.

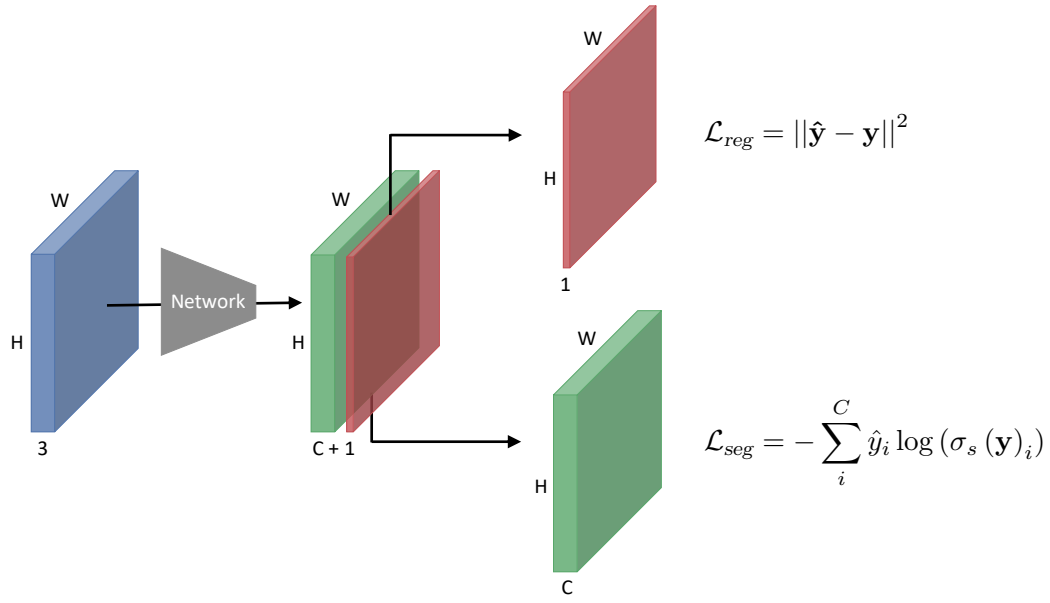


Fig. 5.16 Implementation of multi-task learning by adding an extra output channel to the network and applying separate losses.

5.4.3 Multi-task learning

Multi-task learning is where a network is trained to perform two or more tasks simultaneously, i.e. its loss function consists of an element for each task. The network receives training signals from all the tasks and it must learn features which help it perform all tasks well. By doing this, the network is able generalise better on the original task [46, 187].

In this thesis, multi-task learning is proposed as a mechanism for introducing domain-specific knowledge into the learning. As already mentioned in Chapter 4, the horizon is a key feature of the maritime landscape and could provide useful information, for example, that sea regions should be below the horizon line and sky regions above. One of the challenges of semantic segmentation is to achieve precise boundaries between neighbouring semantic regions, especially when there is low visual distinction between them. As noted in previous chapters, this is often the case with maritime objects (e.g. low contrast targets in the distance, or light-coloured objects causing wake). Two multi-task set-ups were therefore investigated: one where the network must predict the horizon, as well as the class segmentation, and one where it must predict the boundaries between semantic regions.

The EDANet architecture was modified to produce predictions for the secondary task. No new convolutional layers were added so that the number of features being learned was

the same for comparison with previous experiments. The objective is to test if the network learns *better* features, not more of them. The modifications were therefore restricted to the minimum that would allow the multi-task loss to be applied to the outputs.

The additional tasks were modelled as an additional channel in the output tensor, such that the new number of output channels for C classes was $C+1$. This is implemented as an additional channel in the final projection layer (1×1 convolutions) before the final upsampling step. During training, the additional output channel was separated and the cross-entropy classification loss and secondary task regression loss were applied separately. This is shown in Fig. 5.16.

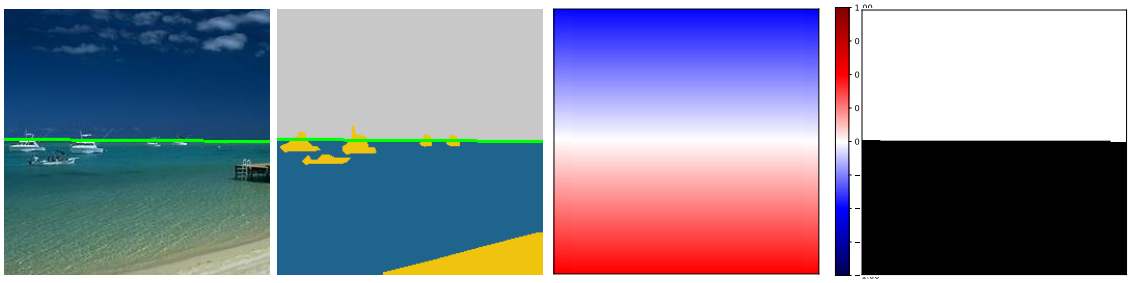
Predicting distance to horizon

In this task, the network is trained to predict the vertical distance to the horizon line for each pixel. That is, the network must produce a distance map where each pixel value represents the distance (in normalised image coordinates) from the horizon, above or below it. The pixel values are therefore in the range $(+1, -1)$, where positive values indicate that the pixel is above the horizon, and negative values are below. This approach was inspired by recent work in the field of instance segmentation [49, 132, 214], where each pixel predicts its offset from object instance centres, as well as its class. The choice of predicting a distance map instead of predicting line parameters is to encourage the network to learn more about the relative position of classes in the scene (e.g. that sea should be below the horizon, sky should be above, etc.).

For training and evaluation, horizon groundtruth was created for the ADE20k maritime subset and MarSemSeg images. The horizon line was drawn on by hand and the line parameters were saved to a file. At training time, the line parameters are used to generate a horizon map 'on the fly'. Some images did not contain the horizon (e.g. those from downward-looking aerial viewpoints) so these were marked as 'void' and are ignored during calculation of the horizon prediction loss. Fig. 5.18 shows an example horizon map. The horizon line can be easily recovered by binarising the map into regions greater and less than zero.

Predicting location of region boundaries

Semantic region boundaries are different to edges, as edges can occur *within* a semantic class, as well as between them. The choice of predicting semantic boundaries is to

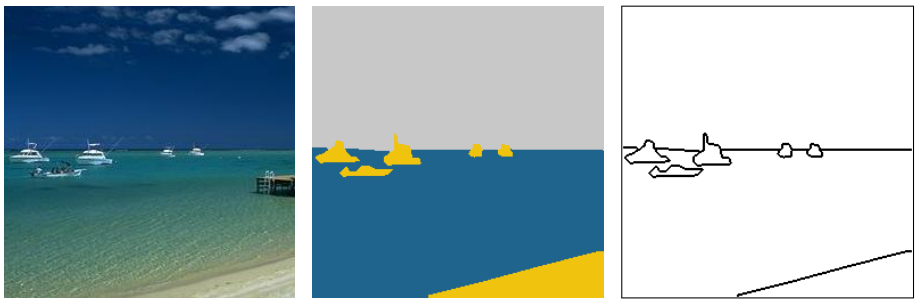


(a) Horizon (b) Segmentation (c) Horizon map (d) Binarised map

Fig. 5.17 Example horizon map examples.

encourage the network to learn features which make the boundaries between different classes clear, even if there is not a strong edge (e.g. due to low contrast). This approach has been taken by [87, 88, 138, 234, 236] to improve boundary accuracy and instance segmentation. In this task, the network is trained to output a binary map for the semantic region boundaries, where 1 indicates that the pixel is on the boundary and 0 means it is not. L2 regression loss was chosen instead of a binary classification loss so that the network outputs a wider range of values between 0 and 1 (rather than encouraging it to predict a polarised bimodal distribution). As semantic boundaries might be ambiguous, a wider range of values could be used as a kind of edge probability map in subsequent processing steps to give more nuanced information about region boundaries.

For training and evaluation, boundary groundtruth was created for the ADE20k maritime subset and MarSemSeg images. This was done automatically by finding the points on the edge of each region in the class label images and saving a boundary label image which is loaded by the network during training. Fig. 5.18 shows an example boundary map.



(a) Image (b) Segmentation (c) Boundary map

Fig. 5.18 Example boundary map.

Multi-task loss function

The loss is made up of two parts: a pixel-wise cross-entropy loss for the segmentation task ($\mathscr{L}_{seg}$) and a pixel-wise L2 regression loss for the secondary task ($\mathscr{L}_{reg}$):

$$\mathscr{L}_{seg} = -\sum_{i}^{C} \hat{y}_i \log\left(\sigma_s\left(\mathbf{y}\right)_i\right) \tag{5.7}$$

$$\mathscr{L}_{reg} = \left|\left|\hat{\mathbf{y}} - \mathbf{y}\right|\right|^2 \tag{5.8}$$

where $\mathbf{y}$ is the network outputs, $\hat{\mathbf{y}}$ is the groundtruth labels, C is the number of classes, and $\sigma_s\left(\mathbf{y}\right)$ is the Softmax function applied to the network output such that:

$$\sigma_s\left(\mathbf{y}\right)_i = \frac{e^{y_i}}{\sum_j^C e^{y_j}} \tag{5.9}$$

One of the challenges in the multi-task learning is how to combine the components of the loss into a single function. A common approach is to take a weighted sum of each loss:

$$\mathscr{L} = \omega_1 \mathscr{L}_1 + \omega_2 \mathscr{L}_2 + \cdots + \omega_n \mathscr{L}_n \tag{5.10}$$

where the values of ω are manually specified weights for each component of the loss. The weights are tuned empirically to achieve good performance. A more principled approach was proposed by Kendall et al. [115] which weights each loss component according to the uncertainty of its task. This is the approach adopted in these experiments so the multi-task loss function is:

$$\mathscr{L} = \frac{1}{\sigma_{seg}^2}\mathscr{L}_{seg} + \log\sigma_{seg} + \frac{1}{2\sigma_{reg}^2}\mathscr{L}_{reg} + \log\sigma_{reg} \tag{5.11}$$

where $\mathscr{L}_{reg}$ is used for both the horizon prediction and boundary prediction loss.

The formulation is based on likelihood maximisation using a softmax likelihood and a Gaussian likelihood to model the classification and regression tasks, respectively. This creates a loss function which is also dependent on the noise parameters for each task, σ_{seg} and σ_{reg}. During training, the network learns the relative weights of the losses by learning values for σ_{seg} and σ_{reg} which minimise the loss. As the noise in one of the tasks increases, its weight decreases, and *vice versa*. At the same time, the $\log\sigma$ terms act as regularisation to stop the noise from increasing too much.

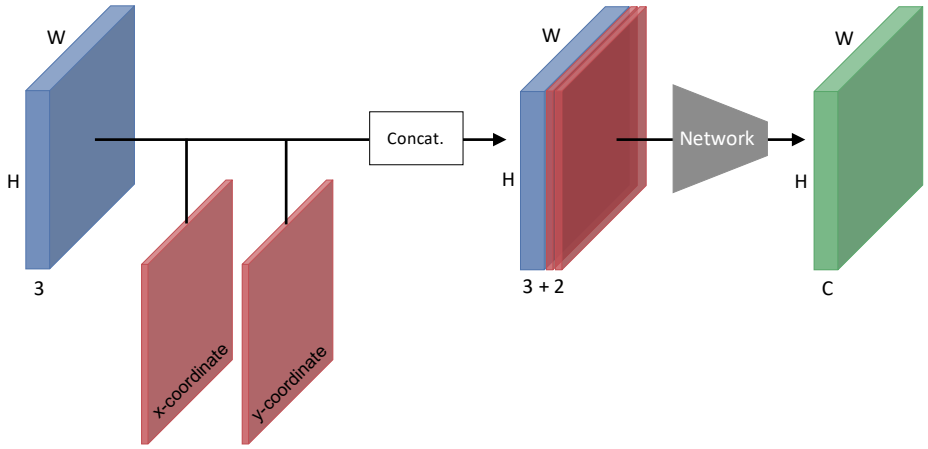


Fig. 5.19 CoordConv concept [137]

In the implementation, the trained variable in the loss function is actually $s = \log\sigma^2$ such that the noise parameters (standard deviations) are $\sigma = (e^s)^{\frac{1}{2}}$ and the loss weights are $\frac{1}{\sigma^2} = e^{-s}$. This is done to avoid potential division by zero and to ensure valid standard deviation values are obtained.

5.4.4 Incorporating global spatial information

By design, convolutional networks are translation-invariant, allowing them to recognise features anywhere in an image. However, not all objects and features are equally likely to occur in every location in an image. This is even more the case in images from a domain such as maritime surveillance where the real world scene makes certain objects and features much more or less likely in different locations.

CoordConv [137] is a way of providing the network with extra information about the global location of a pixel within the image, allowing the network to learn different amounts of translation invariance in its features. It works by providing the network with the (x, y) coordinate of each pixel in the image (in normalised image coordinates). In practice, this is achieved by two extra channels in the input image tensor: one which encodes the x-coordinate and one for the y-coordinate (see Fig. 5.19).

Liu et al. [137] note that the spatial information does not bring much benefit to image classification tasks, as the precise location of features within the image is not important for determining the overall class. However, they have shown it to be useful in object detection (where knowing the location of objects within the image is important) and reinforcement learning (where knowing the global layout of a game can improve strategy

Table 5.6 Key to semantic segmentation method variants (all use the EDANet network)

Variant Name	Augmentation	Multi-task	CoordConv
baseline	-	-	-
aug	✓	-	-
coordconv	-	-	✓
aug-coordconv	✓	-	✓
horizon	-	Horizon	-
horizon-aug	✓	Horizon	-
horizon-coordconv	-	Horizon	✓
horizon-aug-coordconv	✓	Horizon	✓
boundary	-	Boundary	-
boundary-aug	✓	Boundary	-
boundary-coordconv	-	Boundary	✓
boundary-aug-coordconv	✓	Boundary	✓

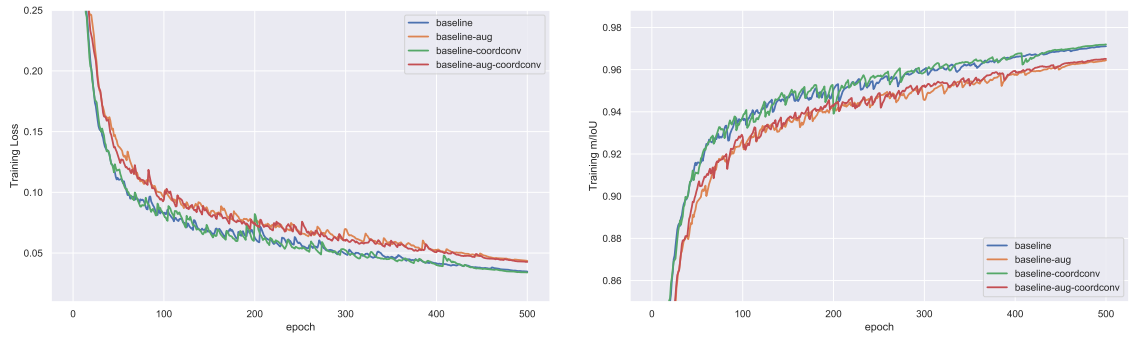
and performance). To the best of the author's knowledge, CoordConv has only been applied to the task of semantic segmentation in one other area – seismic depth image analysis [113] – and so a contribution of this thesis is the first time it has been explored in the context of real-world scenes and maritime surveillance.

5.4.5 Results and analysis

In total, 12 different configurations of EDANet were trained. Details are listed in Table 5.6. For all configurations, the training data and scene parsing step are kept the same, and the training hyperparameters are set as per the original EDANet paper [139]. Training and evaluation are run on the same computer platform as before.

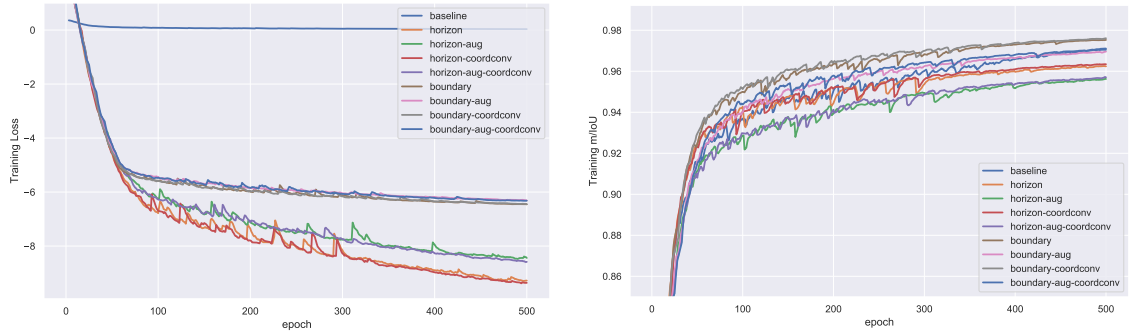
Effect on training

Adding large amounts of data augmentation causes the network to train more slowly and the loss/mIoU doesn't reach as low/high a value (Fig. 5.20). This is as expected, as the network never sees exactly the same image twice. Looking at the curves for the multi-task cases (Fig. 5.21), train mIoU reaches a slightly flatter plateau which could indicate earlier convergence. Applying CoordConv on its own or in combination with any of the other methods does not have an impact on loss or mIoU.



(a) Training loss (b) Training mIoU

Fig. 5.20 Training loss and mIoU for the baseline variants

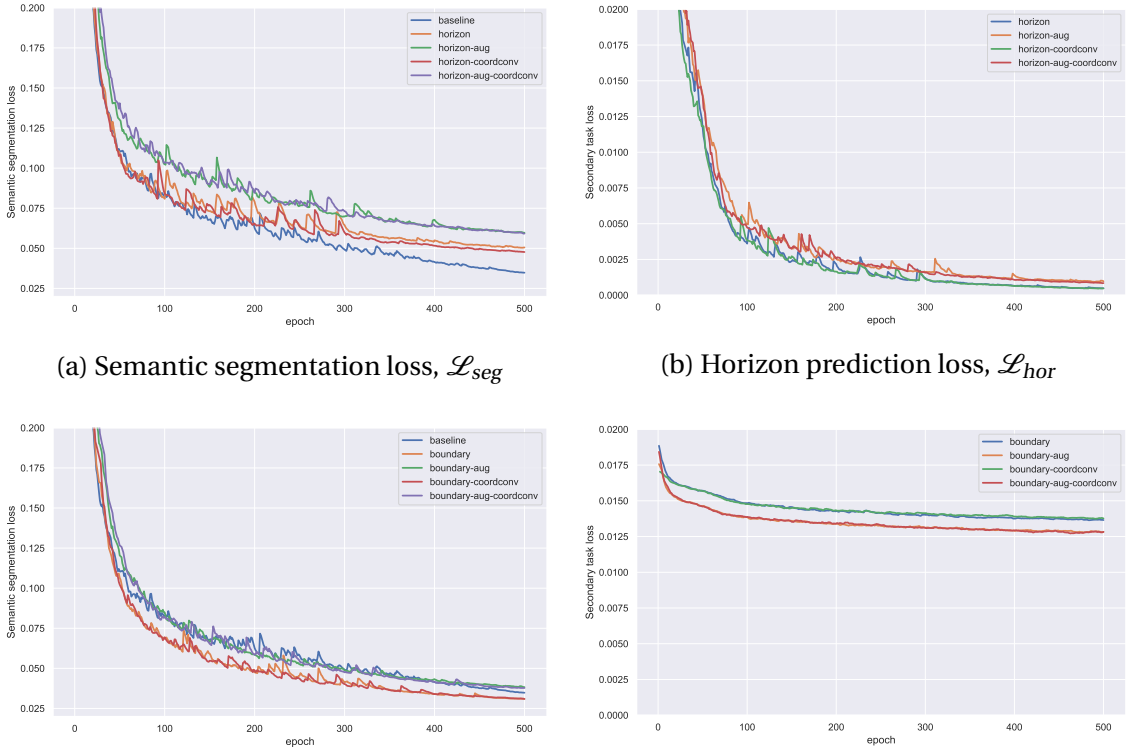


(a) Training loss (b) Training mIoU

Fig. 5.21 Training loss and mIoU for the multi-task variants

For the multi-task variants, the combined training loss (Fig. 5.21a) seems to be better for the horizon prediction task than the boundary prediction task. However, analysis of the component losses in Fig. 5.22 shows that the semantic segmentation loss is lower with boundary prediction. The boundary task also leads to higher training mIoU in Fig. 5.21b and is the only case where it is higher than the baseline. This suggests that, of the two tasks, boundary prediction is adding the most benefit. Note, the negative losses in the multi-task cases are a result of the log operations in Eqn. 5.11.

Finally, an interesting observation is that data augmentation increases the loss for horizon prediction, but decreases it for boundary prediction (Fig. 5.22b and 5.22d). A possible explanation for this is that the invariances learned from the different augmentations are all helpful for the classification task (e.g. invariances to colour and lighting variations) but some of them make the horizon prediction task more difficult (e.g. rotation).



(a) Semantic segmentation loss, $\mathscr{L}_{seg}$

(b) Horizon prediction loss, $\mathscr{L}_{hor}$

(c) Semantic segmentation loss, $\mathscr{L}_{seg}$

(d) Boundary prediction loss, $\mathscr{L}_{bound}$

Fig. 5.22 Training curves for horizon prediction (a-b) and boundary prediction (c-d) multi-task training.

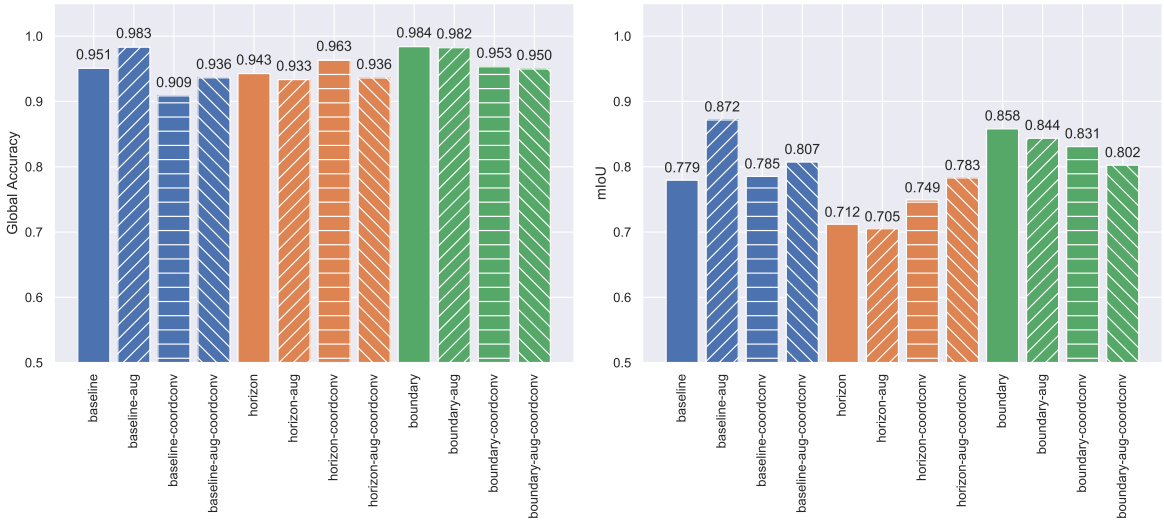


Fig. 5.23 Global accuracy and mIoU results for the semantic segmentation network variants on MarSemSeg

Effect on segmentation performance

The global accuracy and mIoU results in Fig. 5.23 confirm that boundary prediction is adding the most benefit out of the two secondary tasks. A notable observation is that data augmentation on its own is comparable to or outperforms the other configurations. Additionally, augmentation does not have the same impact when combined with horizon or boundary prediction; in fact, there is a slight decrease in score when augmentation is applied. CoordConv also has an inconsistent effect. In the horizon prediction task, it has a positive impact, whilst in the boundary prediction task, it decreases performance (however, boundary prediction with CoordConv is still better than the baseline).

Looking at the per-class scores in Fig. 5.24, the patterns for Sea and Sky are very similar, specifically that augmentation tends to increase performance whilst CoordConv tends to reduce it. However, with the Other class, CoordConv can be observed to have a positive effect. This is an important result, as the Other class is critical for reliably detecting objects.

Qualitative results

Figs. 5.25 - 5.28 show some example outputs from the different network variants. Augmentation visibly improves the output, reducing false positives and producing cleaner object segmentations (Fig. 5.25). As discussed above, CoordConv often has a negative impact. An example of this can be seen in Fig. 5.26. The baseline method creates a good segmentation

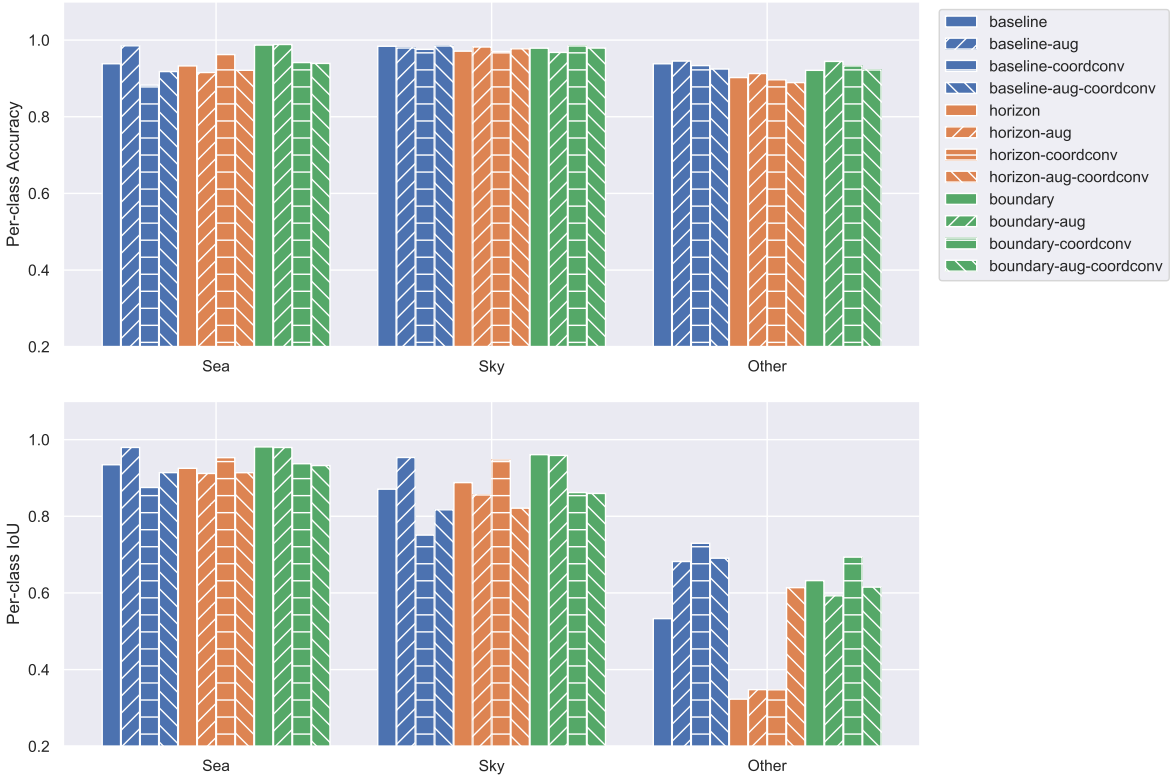
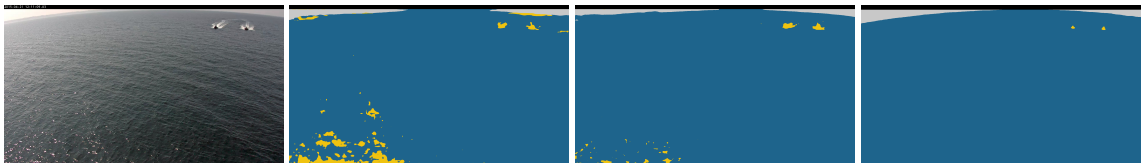
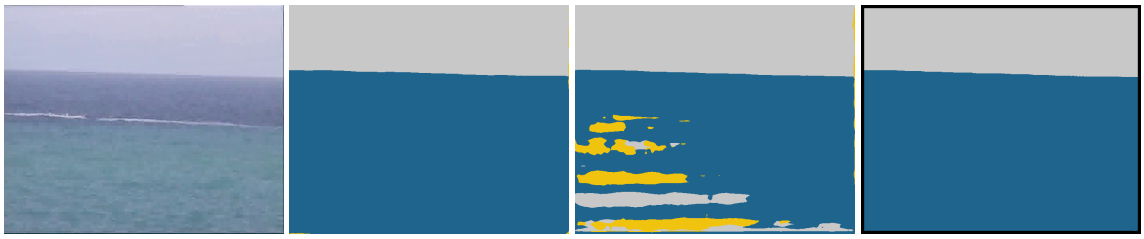


Fig. 5.24 Per-class accuracy and IoU results for the semantic segmentation network variants on MarSemSeg



(a) Image (b) baseline (c) aug (d) Groundtruth

Fig. 5.25 Qualitative results showing effect of data augmentation



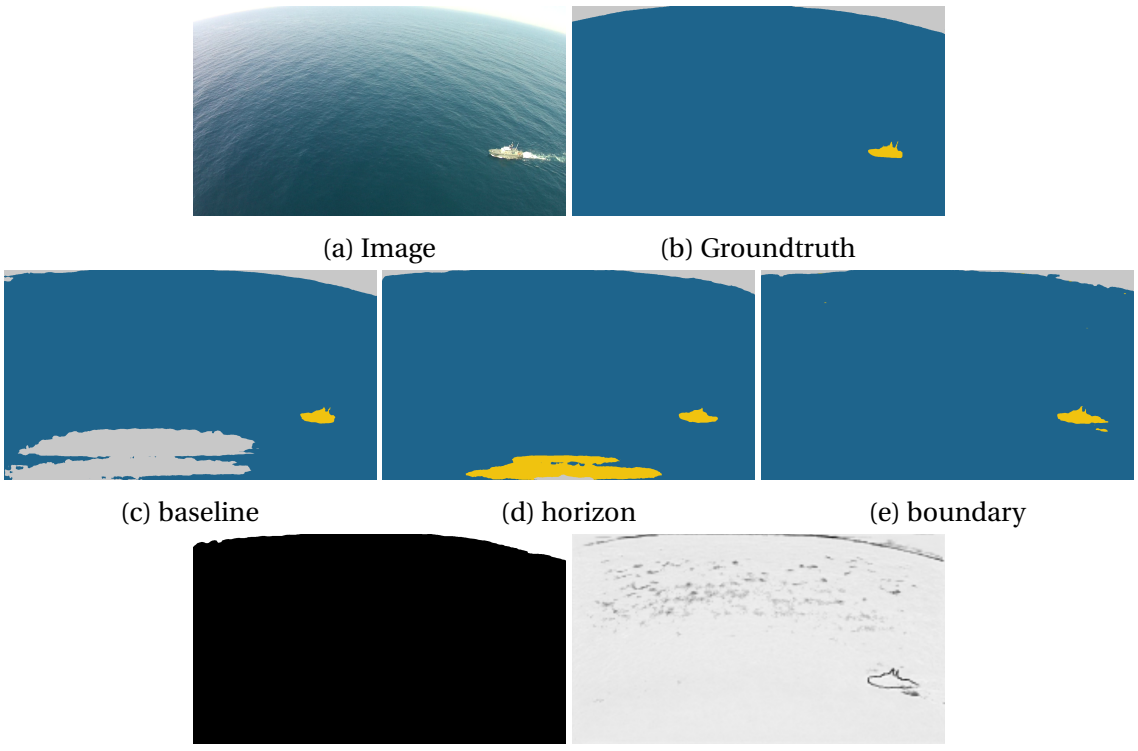
(a) Image (b) baseline (c) coordconv (d) Groundtruth

Fig. 5.26 Qualitative results showing effect of CoordConv

of an easy scene but CoordConv introduces artefacts. It is suspected that this is because the network did not see many images of empty sea and is therefore expecting to see the Other class present at the bottom of the image.

The effect of horizon and boundary prediction is shown in Fig. 5.27 using an example image from an aerial viewpoint. The baseline method incorrectly detects a large sky region in the lower part of the image. With horizon prediction, there is still a false detection but the class is Other, suggesting that the network is aware of relative positions of classes in the image (i.e. that the Other class is much more likely to be found lower in the image than Sky). With boundary prediction, the falsely detected region is eradicated. The boundary map shows that the network is aware that there is no separate region in that part of the image.

Boundary prediction also allows the network to capture more of difficult objects, such as those in Fig. 5.28. The tall, thin mast has been captured well in the boundary map and this has been reflected in the segmentation. With CoordConv applied as well, the result is better still, suggesting that it supports capture of more complete boundaries.

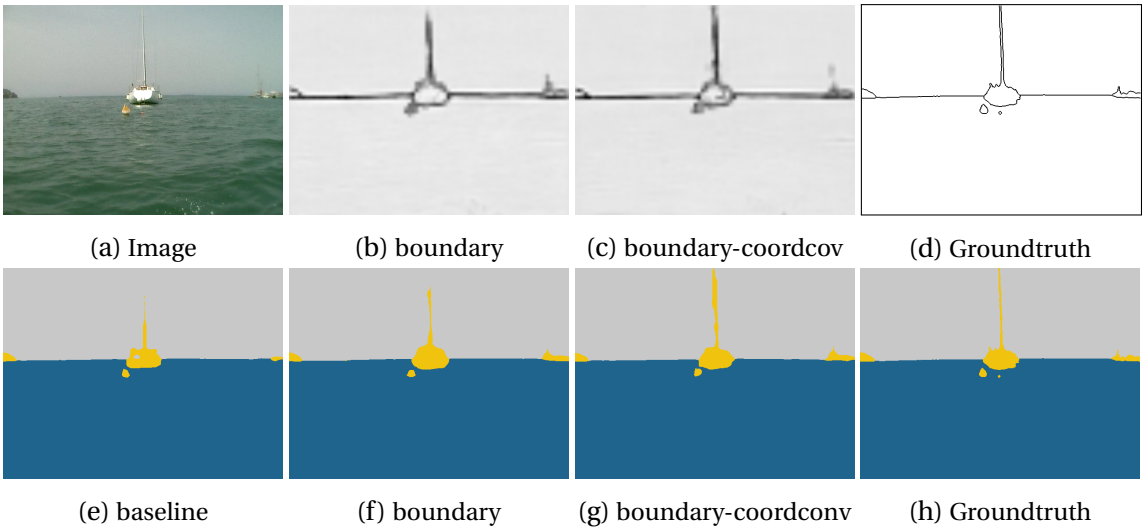


(a) Image (b) Groundtruth

(c) baseline (d) horizon (e) boundary

(f) Horizon map (g) Boundary map

Fig. 5.27 Qualitative results showing effect of the horizon prediction and boundary prediction tasks



(a) Image (b) boundary (c) boundary-coordcov (d) Groundtruth

(e) baseline (f) boundary (g) boundary-coordconv (h) Groundtruth

Fig. 5.28 Qualitative results showing effect of the boundary prediction task and CoordConv

5.5 Creating an object detector for maritime surveillance

In this chapter, object detection is framed as the task of finding regions of the scene which are “not sea or sky”. The role of the semantic segmentation network is to map out the sea and sky regions as accurately as possible and identify regions which do not fall into one of these classes. The next step in the proposed approach is a reasoning process which uses constraints relating to maritime scenes to parse the network output and identify maritime objects. A diagram of the proposed approach is shown in Fig. 5.29.

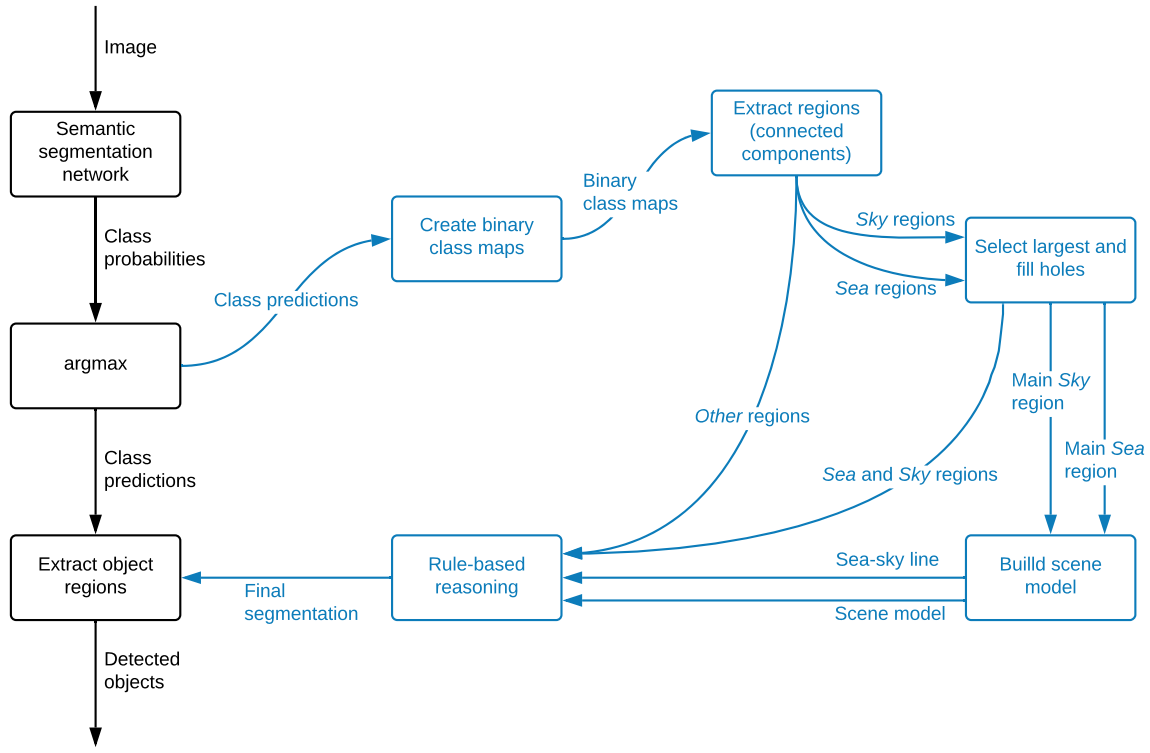


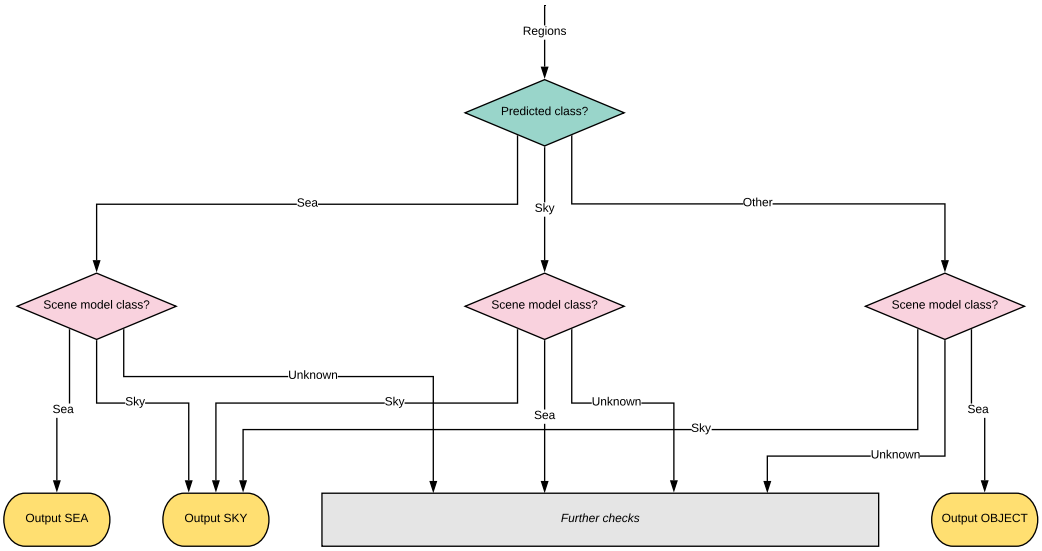
Fig. 5.29 The proposed semantic segmentation-based object detection method. The parts in black show the ‘naïve’ approach; the parts in blue show the extended approach which applies domain knowledge to extract objects more effectively.

The image is passed through the semantic segmentation network to generate a probability distribution over classes (Sea, Sky and Other) for each pixel. The maximum probability determines the network’s predicted class for each pixel. A naïve way to extract objects at this point would be to simply output all the Other regions as object detections. However, this would not account for false positives and would not be able to distinguish between

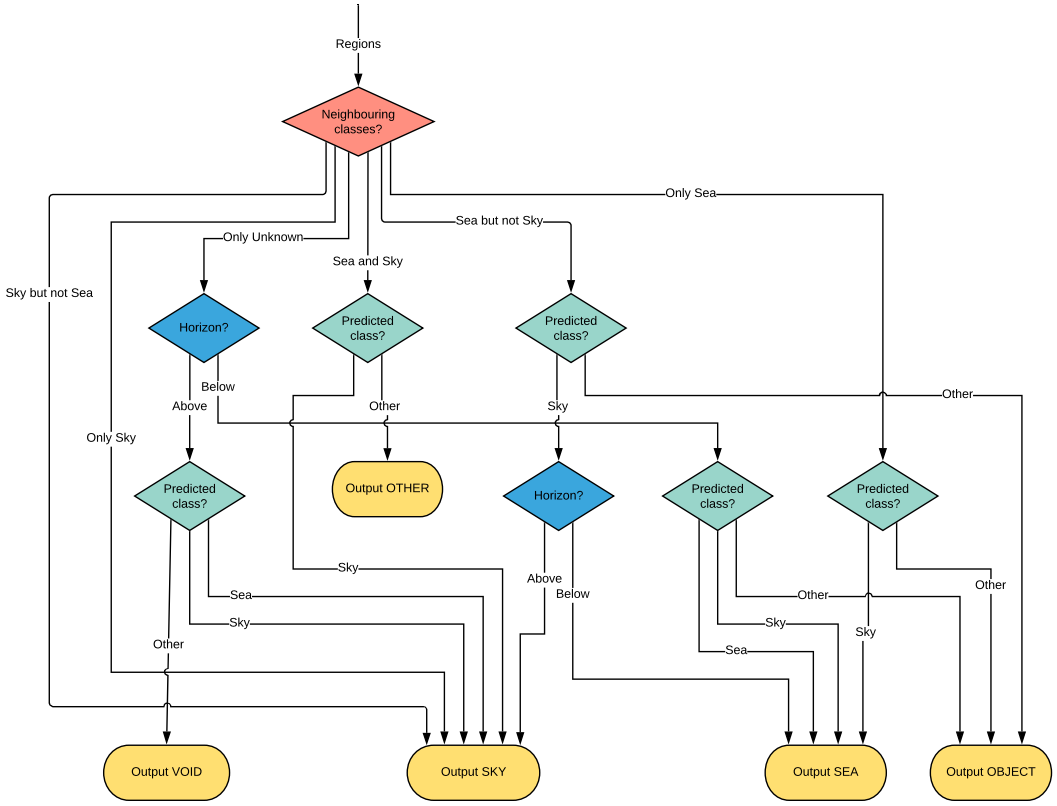
objects and land. Instead, the class predictions are first used to create a binary map for each class from which regions can be extracted by analysing connected components.

The largest regions for sea and sky are identified and holes are filled. These are then combined to form a model of the scene which consists of sea, sky and some unknown regions. The sea-sky line is also computed by finding the set of points where the sea and sky region meet. A set of simple rules and constraints is then used to process the remaining Sea and Sky regions, along with the Other regions from the previous step. The reasoning process (depicted in Fig. 5.30) compares the predicted class of a region with its own class and neighbouring classes in the scene model to determine if it is an object, land or a false positive. If it is a false positive, the reasoner tries to resolve what the true class is to reduce false detections. The sea-sky line is also used as part of this process.

The output of the reasoning step is a revised class segmentation which distinguishes Objects as well as the 3 original classes of Sea, Sky and Other (see Fig. 5.31 for examples). Detections are output in the form of bounding boxes, based on the regions in the segmentation. Depending on the requirements, it could be useful to output Land bounding boxes too, but in this work, only Object bounding boxes are output from the detection module.



(a) Initial checks

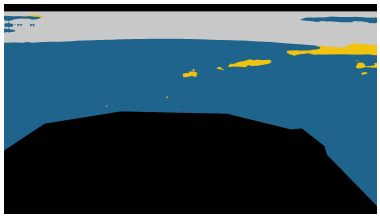


(b) Further checks

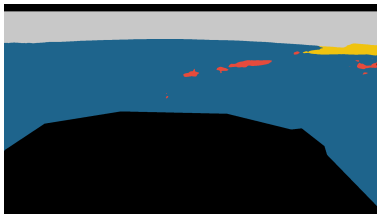
Fig. 5.30 Scene reasoning decision process. In the *Initial Checks* (a), the predicted class of a region is compared against its own class from the scene model. If this does not resolve the output class, *Further Checks* (b) are performed which use neighbouring classes in the scene model and the horizon to resolve the output class.



(a) IPATCH 2015 Sc2a_Tk1-CAM12



(b) Segmentation from argmax



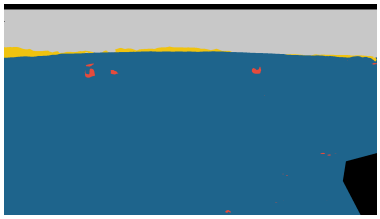
(c) Segmentation after reasoning



(d) IPATCH 2017 Sc3a-CAM12



(e) Segmentation from argmax



(f) Segmentation after reasoning

Fig. 5.31 Output of the scene reasoning decision process compared to argmax class predictions. Compared to (b) and (e), (c) and (f) distinguish the boats as Objects and the land in the distance as Other. Falsely classified regions in the sky are also corrected.

5.6 Evaluation and comparison against baselines

5.6.1 Experimental set-up

Sequences and metrics

The sequences used for evaluation are listed in Table 3.3. The MODP-BEP3, Detection Rate and FAF metrics are used for quantitative analysis, as described in Chapter 3. In addition, to assess real-time performance, the processing speed of the proposed method is measured for each frame.

Implementation and configurations

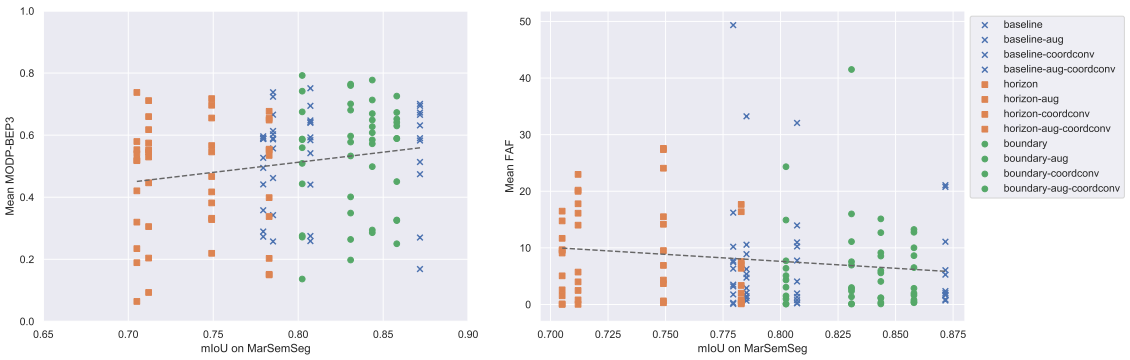
The proposed object detection system is implemented in Python to integrate easily with the trained models in the PyTorch framework. Twelve different object detection configurations are evaluated based on the twelve different trained models listed in Table 5.6. Training and object detection are run on the same Alienware laptop with an 8-core 2.6GHz Intel® Core™ i7 CPU and 16GB RAM, with an externally connected NVIDIA® GeForce® GTX™ Titan X GPU with 12GB memory.

5.6.2 Results and analysis

It would be expected for the object detection performance to correlate with segmentation performance. To test this, the mean MODP-BEP3 scores for each sequence in Table 3.3 are plotted against the mIoU scores achieved on MarSemSeg for each network variant. This is shown in Fig. 5.32. There is a slight correlation for both MODP-BEP3 and FAF (higher MODP-BEP3 is better, lower FAF is better), but it is not significant. For a given trained model, there is still large variation in object detection performance across the sequences (see Fig. 5.33 - 5.35). However, even with very limited training data, the ability to generalise to different viewpoints, environmental conditions and object types is promising.

Effect of multi-task learning

In general, the boundary prediction task supports the task of object detection more than training with horizon prediction or just training for segmentation on its own. A possible explanation for this is that there is a lot of variation in where the Other class occurs in the training images, so knowing the relative position of the horizon does not provide a



(a) MODP-BEP3 vs. mIoU

(b) FAF vs. mIoU

Fig. 5.32 Relationship between segmentation performance and object detection performance.

lot of information. On the other hand, discerning boundaries well is an important part of segmentation, so additional training signals from the boundary error helps the network maximise its IoU score which is also beneficial for object detection.

In the SMD sequences (Fig. 5.34), there is much less variability of performance across the sequences and methods (all distributed around ~0.6 MODP-BEP3) compared to IPATCH (Fig. 5.33) and SEAGULL (Fig. 5.35). This is partly due to SMD containing shorter sequences with stationary objects so the detection challenges do not vary much. The best method across all SMD sequences is baseline-aug (only data augmentation used). This could suggest that the SMD sequences are the closest to the training data and that adding extra tasks like horizon and boundary prediction weakens the classification ability of the features that the network learns.

Effect of augmentation and CoordConv

Neither augmentation nor CoordConv have a consistently positive or negative effect over configurations and sequences. At the same time, their impact is not negligible; in some cases, adding augmentation or CoordConv can dramatically change the performance. For example, adding augmentation has a large negative effect in IPATCH 2017-Sc6b-CAM10 but a positive effect in IPATCH 2016-Sc1_Tk5-CAM11 (see Fig. 5.33). CoordConv tends to have a positive effect on most IPATCH sequences when used on its own, but the impact is reduced (or becomes negative) when augmentation is applied at the same time. It is not clear why augmentation and CoordConv interact with performance in such unpredictable ways. It's possible that these mechanisms are causing the network to overfit on the training

data in ways which were not apparent during training, rather than helping to generalise, leading to highly data-dependent results.

Qualitative results and limitations

The proposed semantic segmentation-based approach is able to detect challenging targets in some sequences with some of the trained models. For example, in IPATCH 2016-Sc1_Tk5-CAM11 (Fig. 5.36), the baseline-coordconv variant detects the two small targets consistently throughout the sequence and with very few false positives. In the SMD sequences, the baseline-aug model correctly detects even smaller and fainter targets (Fig. 5.37a and 5.37b), however it does not detect a much larger object (albeit a low contrast one) in Fig. 5.37c. It also suffers from the same problem as the saliency-based approach of merging targets if they are overlapping (Fig. 5.37d).

In other sequences and with other variants, many more false positives are detected and the localisation is not as precise (Fig. 5.38). Reflections and glare remain a challenge and wake is detected. Detecting wake can assist detection for distant targets (e.g. Figs. 5.38a and 5.38d) but degrades performance when the objects are closer (e.g. Figs. 5.38b, c, e and f). The training data did not contain many images of boats in motion so there was likely not sufficient representation of wake in the training data. Augmentation and multi-task learning are not able to compensate for this.

In Fig. 5.38b, the baseline model detects large regions of wake in the IPATCH 2015-Sc2a_Tk1-CAM12 sequence. When CoordConv is applied (Fig. 5.38e), the boat regions are more precisely localised. However, the same is not true in IPATCH 2017-Sc3a-CAM12 (Figs. 5.38c and 5.38f). The wake regions are more fragmented, but the targets are not necessarily better detected. This is likely due to the white boats in the 2017 sequence appearing very similar to the wake regions.

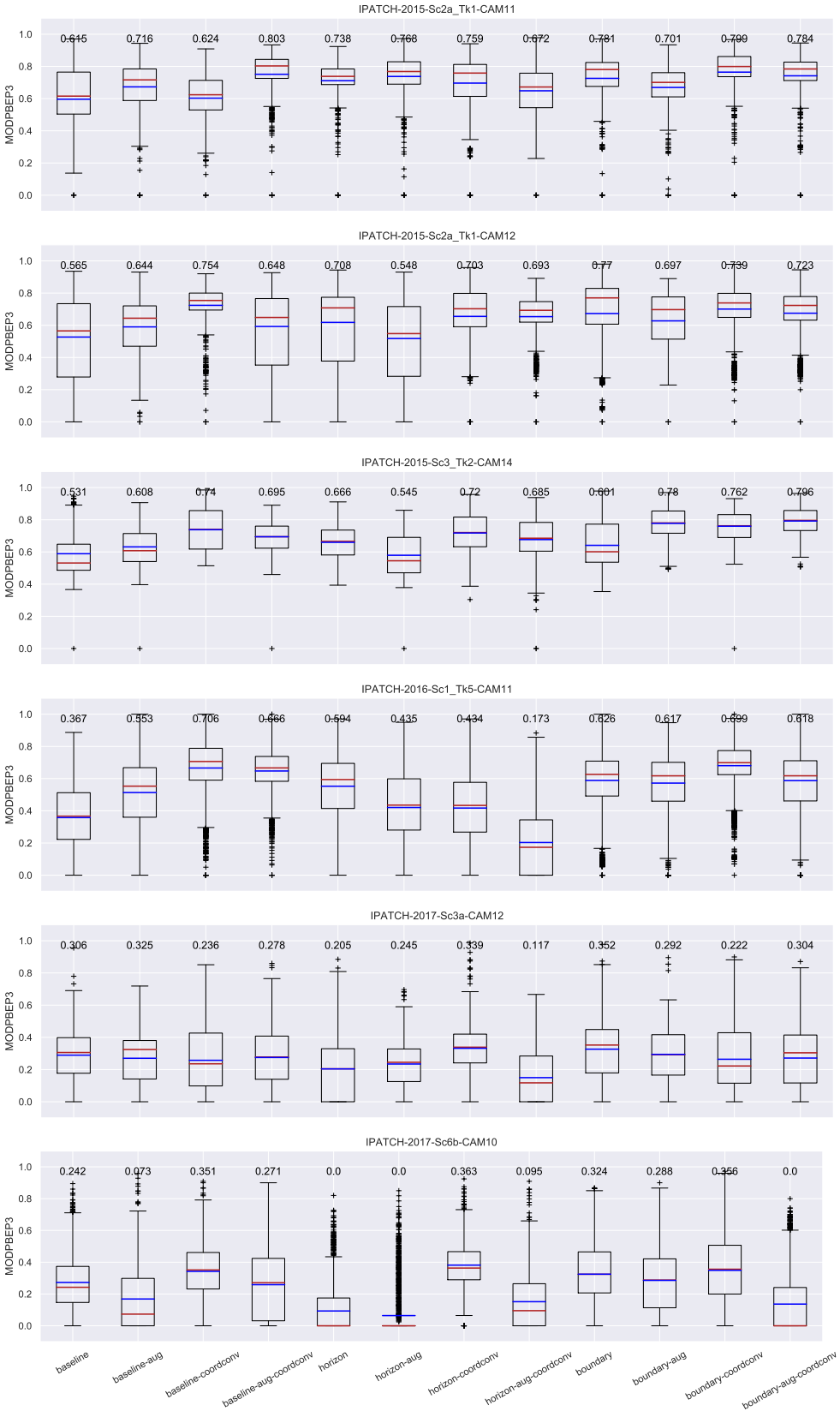


Fig. 5.33 MODP-BEP3 results for the semantic segmentation-based object detection method on the IPATCH sequences.

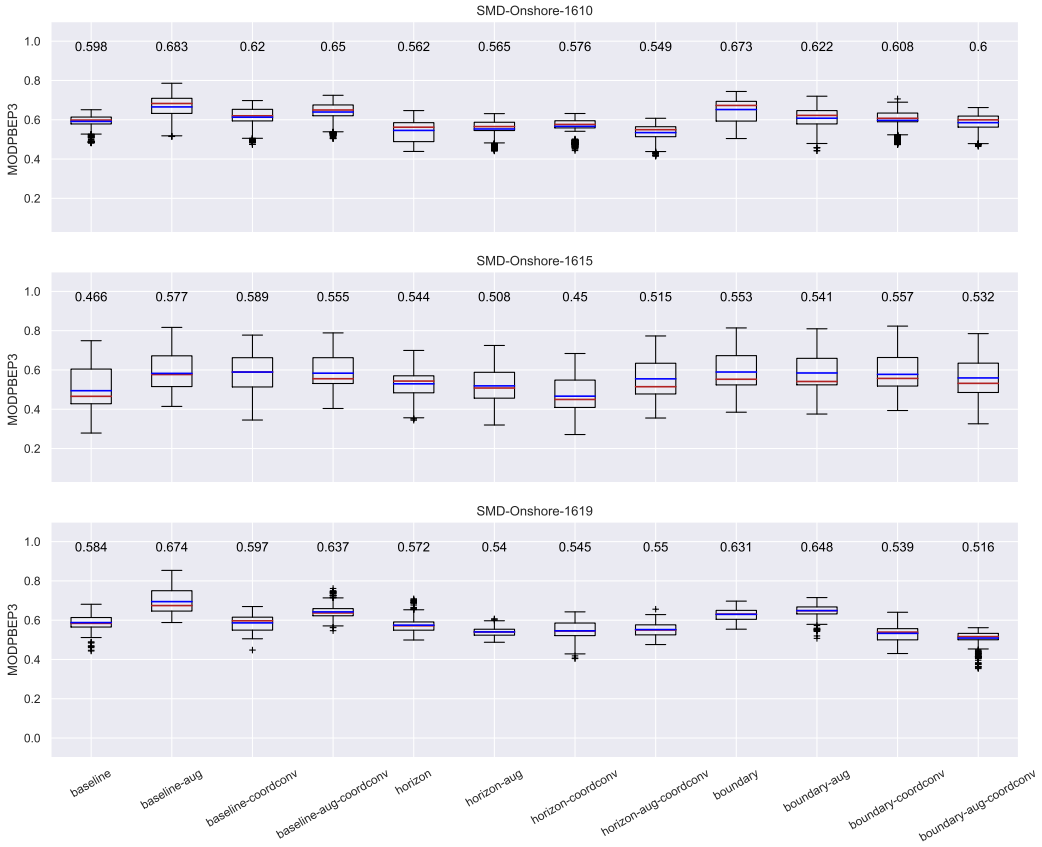


Fig. 5.34 MODP-BEP3 results for the semantic segmentation-based object detection method on the SMD sequences.

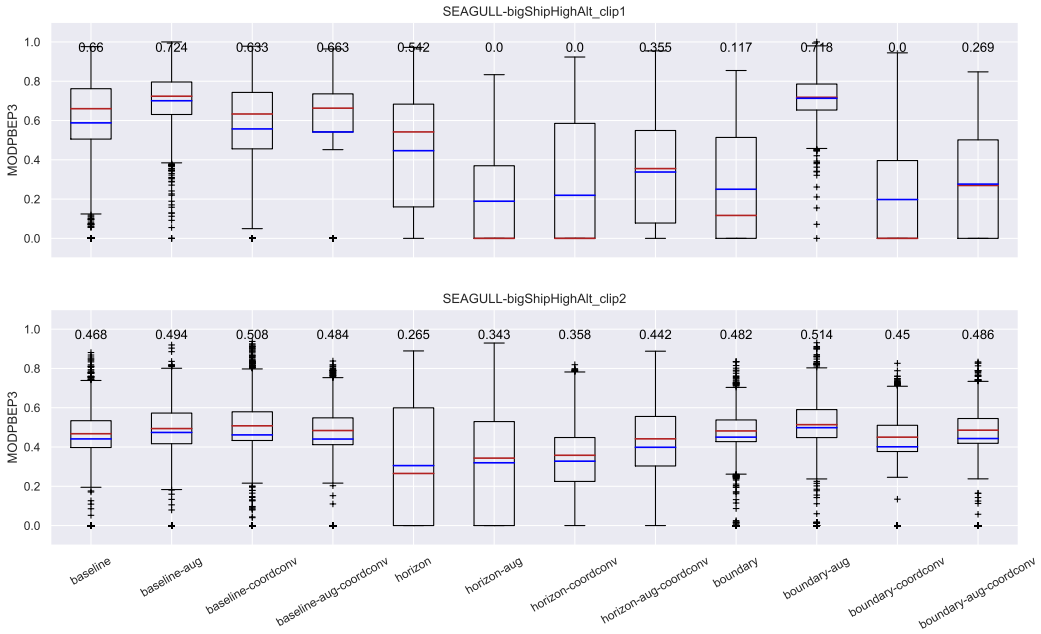
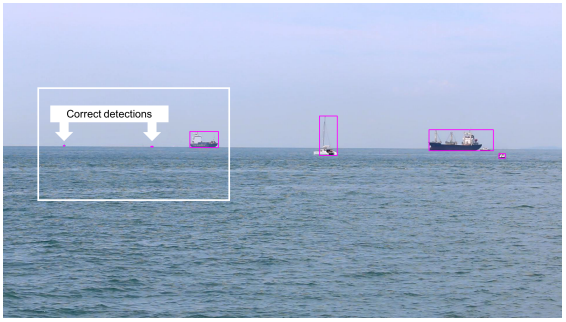


Fig. 5.35 MODP-BEP3 results for the semantic segmentation-based object detection method on the SEAGULL sequences.



(a) Frame 1600 (b) Frame 2200 (c) Frame 2500

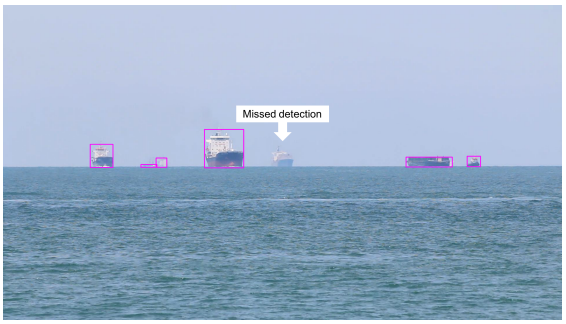
Fig. 5.36 Example of consistent good performance on challenging targets (baseline-coordconv on IPATCH 2016-Sc1_Tk5-CAM11)



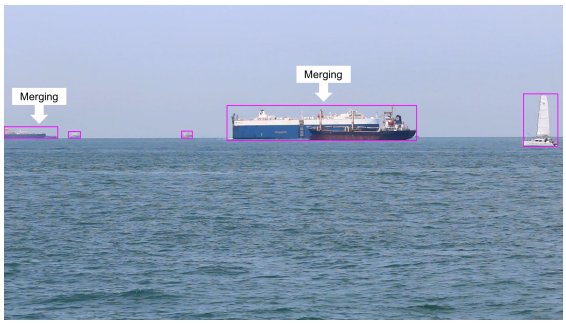
(a) Challenging targets detected in SMD-1610



(b) Zoomed view of frame in (a)

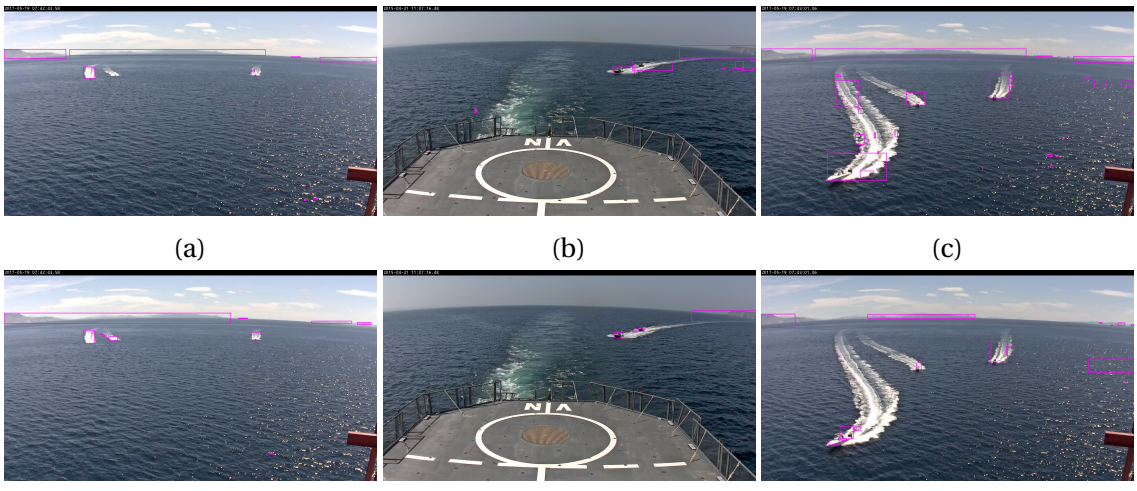


(c) Less challenging targets missed in SMD-Onshore-1619



(d) Merging of overlapping objects in SMD-Onshore-1615

Fig. 5.37 Examples of good performance and limitations on SMD sequences.



(a) (b) (c)

(d) (e) (f)

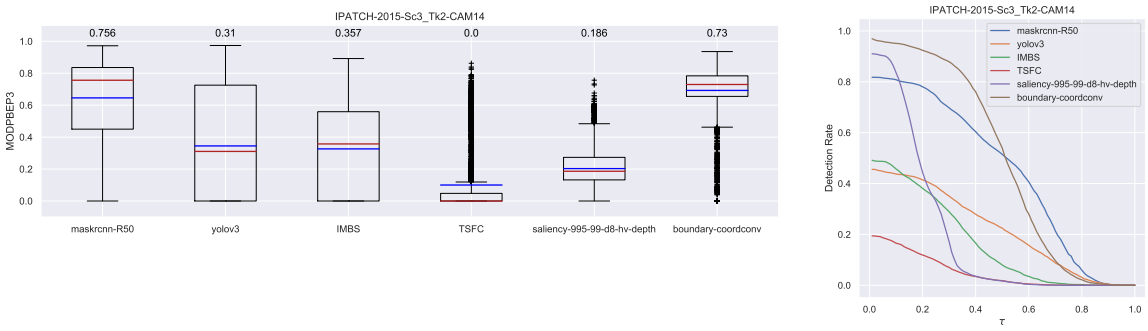
Fig. 5.38 Examples comparing horizon and boundary prediction (a, d) and CoordConv (b, c, e, f) on IPATCH sequences.

Table 5.7 False positive results (FAF) for the proposed and baseline methods on representative IPATCH sequences.

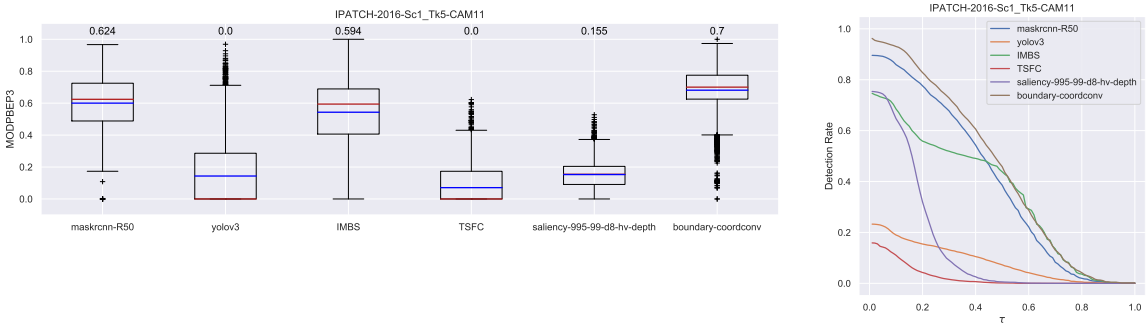
Sequence	maskrcnn-R50	yolov3	TSFC	saliency	semantic
IPATCH-2015-Sc3_Tk2-CAM14	**0.15**	**0.08**	0.55	10.43	41.05
IPATCH-2016-Sc1_Tk5-CAM11	**0.32**	**0.17**	1.13	4.83	2.57
IPATCH-2017-Sc3a-CAM12	**0.02**	**0.07**	6.05	9.67	18.27

Comparison against baselines

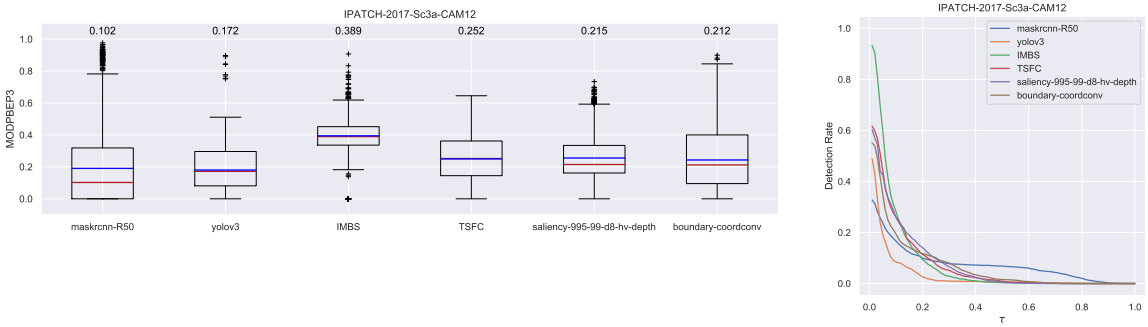
The boundary-coordconv variant is chosen to compare against the baselines and saliency method as it achieves high MODP-BEP3 scores over all the IPATCH sequences. The same three sequences are used as in the previous chapter, as they represent the key challenge in the piracy use case to detect small/distant targets as early as possible. Fig. 5.39 presents the results. Looking at the MODP-BEP3 distributions, the semantic segmentation-based method proposed in this chapter achieves competitive performance compared to the other methods. Its detection rate at low thresholds is also high, which is important for detecting potential threats early, even if the objects are not precisely localised. Unlike the proposed saliency method, the detection rate is maintained over higher thresholds, suggesting that as targets get closer/bigger, localisation accuracy improves. The semantic segmentation method does not perform as well in terms of false positives (see Table 5.7), especially compared to the other deep learning methods (Mask R-CNN and YOLO).



(a) IPATCH 2015-Sc3_Tk2-CAM14



(b) IPATCH 2016-Sc1_Tk5-CAM11



(c) IPATCH 2017-Sc3a-CAM12

Fig. 5.39 Comparison of the proposed semantic segmentation-based object detection method with the saliency-based method from Chapter 4 and baseline methods from the literature.

5.7 Summary

In this chapter, the concept of using deep semantic segmentation networks for object detection was investigated. Based on an initial evaluation of 7 semantic segmentation networks from the literature using the CamVid [41] urban driving benchmark dataset, 4 networks were chosen for further experiments. For training, a subset of the ADE20k [248] scene segmentation dataset was created which contained 434 images containing maritime classes (Sea, Sky, Object, Land and Other). Preliminary tests showed that using 3 classes (Sea, Sky and Other) would be most useful for detecting objects. The preliminary tests also indicated that EDANet [139] was a good balance of accuracy and speed and was therefore selected for more detailed experiments.

To maximise the available training data and test the ability to generalise to the maritime surveillance domain, a new semantic segmentation dataset (MarSemSeg) was created using images from maritime surveillance sequences (see Table. 4.1). MarSemSeg was used to evaluate the performance of EDANet after training with different configurations. Data augmentation was used to provide domain expertise to teach the network about invariances and an ablation study was conducted to analyse the effect of different augmentations. Multi-task learning of horizon and semantic boundary prediction was used to provide the network with additional signals during training. Global spatial information was provided to the network by using the CoordConv approach [137].

To extract objects from the class probabilities output by the semantic segmentation network, a novel rule-based scene reasoning process was proposed. This compares the predicted class of regions with its neighbouring regions and applies spatial relationship rules to resolve misclassified regions and reduce false positives. The proposed method was shown to be competitive against the saliency-based method from Chapter 4 and the baseline methods described in Chapter 3. In the next chapter, the two proposed approaches are compared against the baseline methods from the literature in the context of a real-world piracy surveillance system. Object detections from the methods will be fed into a multi-target tracking module to evaluate how well each method supports the rest of the surveillance pipeline.

Chapter 6

Real-World Performance Evaluation

6.1 Introduction

This chapter investigates the performance of six object detection methods in the context of a complete maritime surveillance system using data collected in the field. The case study is a piracy detection system which was developed as part of the EU-funded research project, 'IPATCH'[1]. The methods under analysis are those from the three publications on which this thesis is based: Temporally-Stable Feature Clusters [156], the saliency-based method from Chapter 4 [44] and the semantic segmentation-based method from Chapter 5 [45]. These are also compared against the baseline methods from the literature: IMBS [30], YOLO [174] and Mask R-CNN [89].

In Chapter 3, 4 and 5, the methods were evaluated using 2D image-based performance metrics. In this chapter, the tracking performance of the system as a whole is evaluated in the real world 3D coordinate system. This gives an assessment of performance which is more relevant to the end users of a maritime surveillance system. The objective is to investigate how different image-based performance characteristics of object detection methods relate to how useful their detections are to the later stages of the surveillance pipeline (tracking, situational awareness, threat recognition, etc.). For example, can false positive detections be eliminated through the tracking process, provided that targets are detected accurately?

First, the on-board surveillance system is described to explain how detections from the object detection methods get processed by other modules of the system. The Multi-Target

[1] IPATCH: Intelligent Piracy Avoidance using Threat detection and Countermeasure Heuristics, www.ipatchproject.eu

Tracking module is explained in more detail, as this is the key process for converting image-based detections into tracks in the real world which can be analysed for suspicious behaviours or threats. The real-world set-up is then described to explain how data was collected during several trials in the IPATCH project on-board real vessels at sea. For the experiments in this thesis, data was used from the trials but the analysis took place offline. The experimental set-up that was used is explained, along with the performance evaluation procedure and metrics. Finally, the results are presented and analysed.

This chapter is based on collaborative work, so individual contribution is clarified where necessary and appropriate reference is made where software or algorithmic components from others have been used.

6.2 On-board surveillance system

6.2.1 Overview

The IPATCH project [102] developed a prototype for an on-board surveillance system to provide situational awareness and alert crews in case of potential piracy threats. The system was designed as a set of modules, each of which fulfilled a role in the surveillance pipeline (Fig. 1.2). Fig. 6.1 shows the high-level architecture of the on-board system. Vessel bridge systems are already equipped with several sensors which can be exploited by the IPATCH system. Under maritime regulations, ships must be equipped with radar, AIS, GPS and an inertial measurement unit (IMU) which measures roll, pitch and heading (yaw), as well as sway, surge and heave. Visual and thermal cameras were added to the vessel to enhance the surveillance capabilities.

The **Early Detection Module** processes data from all available sensors and produces a set of tracks for all detected objects. The **Behaviour Analysis Module** analyses the tracks and enhances them with extra features, such as 'turning', 'accelerating', 'group splitting'. The **Situation Assessment Module** analyses the tracks and features in the context of environmental information, such as weather, details of the vessel, nearby countries, time of year, and so on. This information comes from a series of databases and data feeds. Based on the current situation and detected tracks, the **Threat Detection Module** assesses the threat of pirate attacks or other suspicious behaviour. If the level of threat is high, the crew are alerted through the **User Interface**. If a piracy incident is imminent or in progress, the crew can use the User Interface to access knowledge on countermeasures. The most

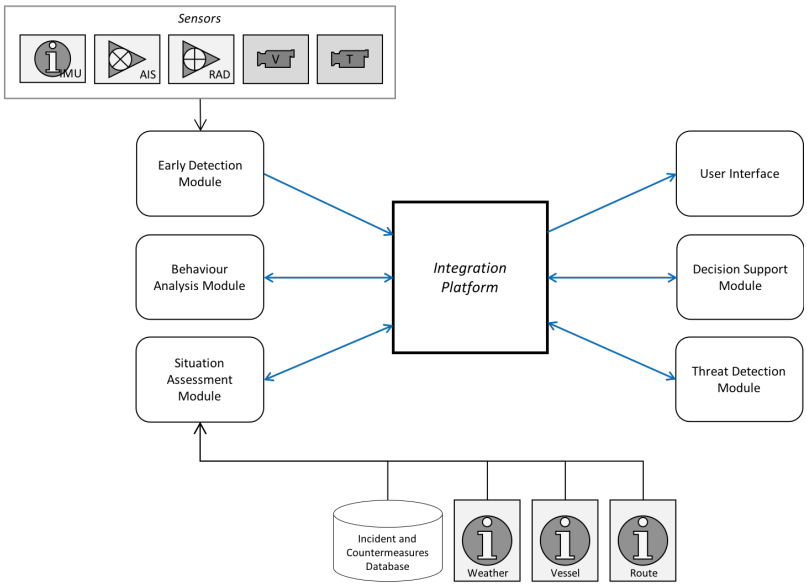


Fig. 6.1 High-level architecture of the IPATCH on-board surveillance system. Key to sensors: IMU = navigational (GPS and intertial measurement unit), AIS = Automatic Identification System, RAD = Radar, V = Visual cameras, T = Thermal cameras)

relevant information and guidance is organised by the **Decision Support System**. Finally, the **Integration Platform** provides the communications infrastructure to pass data and messages between the various modules of the system.

6.2.2 Early Detection Module

The focus of this thesis is on object detection in visual cameras. This process takes place inside the Early Detection Module. Fig. 6.2 shows the components of the Early Detection Module in more detail. A **Bridge Sensor Manager** was developed to connect to the vessel bridge system to collect navigational, AIS and radar data and convert it to a format which can be processed by the rest of the system. Similarly, **Camera Sensor Managers** were developed to ingest raw video feeds from the visual and thermal cameras. The video streams are processed by object detection and tracking methods which operate in image space. The detections are passed to the **Multi-Target Tracking** module which fuses and filters them to produce a consolidated set of tracks which are distributed to other modules for higher-level processing.

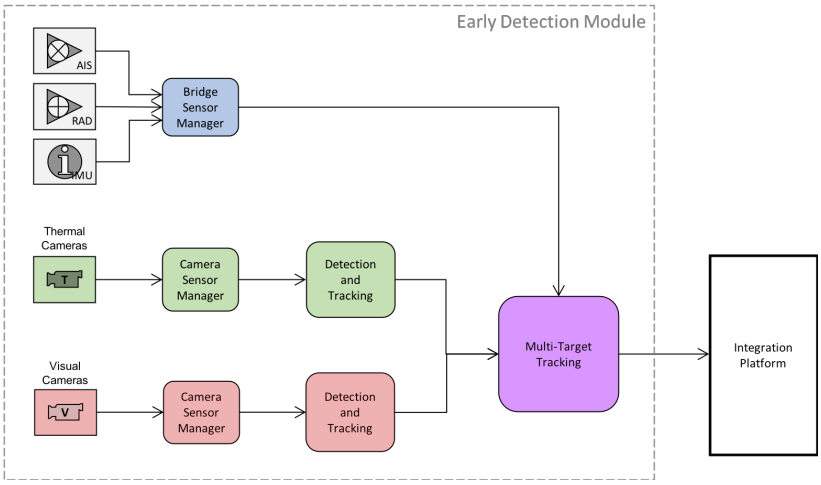


Fig. 6.2 High-level architecture of the Early Detection Module

6.2.3 Multi-target tracking module

The multi-sensor multi-target tracking (MTT) module [4] (see Fig. 6.3) receives detections from the different detection modules. It converts them to ship-centred coordinates, fuses and filters them, and outputs tracks for each estimated target. It receives detections in an asynchronous manner (due to the different processing times and frequencies of each object detection module), so it must first sort and synchronise the inputs over a time window. This introduces a slight delay to allow detections from all modules to arrive before performing fusion and tracking. This delay is adaptive (up to a limit) to adjust for different processing speeds.

The MTT is based on the classical single-hypothesis multi-target tracker from the literature [17, 20]. The position of each track is predicted and filtered using an Extended Kalman Filter and projected into sensor coordinates. A global nearest neighbour algorithm is used to associate detections and tracks and gating is applied so that detections and predictions are only associated if they are close enough. The tracks are then updated with an associated detection or the prediction. Each track is categorised as 'possible', 'active' or 'lost', depending on a number of tunable parameters. For example, if a new track has been detected a certain number of times within a certain time frame, it transitions from 'possible' to 'active'. The reverse happens if the track has not been detected for a certain amount of time, or it may go to 'lost' if longer. Only active tracks are output by the MTT.

The object detection modules produce detections in the form $\begin{bmatrix} x & y & w & h \end{bmatrix}^T$, where x and y are the coordinates of the centre of the *base* of the bounding box, and w and h

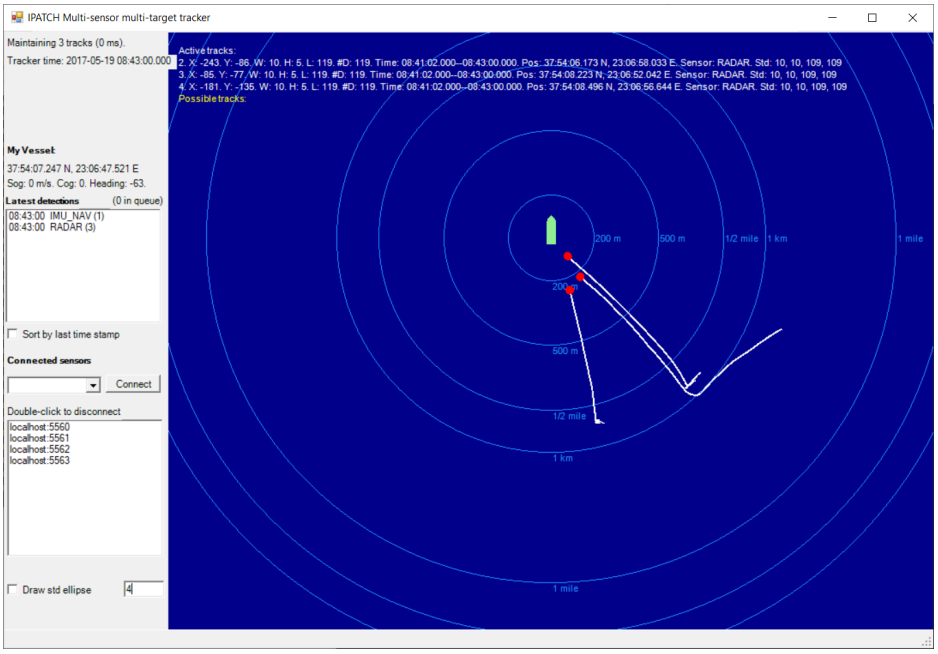


Fig. 6.3 Screenshot of the multi-target tracker (MTT) [4]

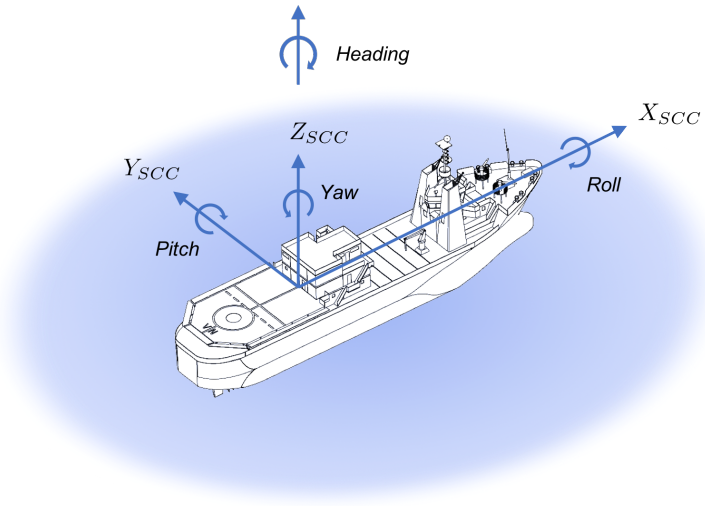


Fig. 6.4 Ship-centred coordinates (SCC)

are the width and height. These are measured in pixels in *rectified* image coordinates. The MTT performs tracking in ship-centred coordinates (SCC). This is a moving frame of reference aligned with the ship's major axes (Fig. 6.4) such that the origin is at the ship's GPS lat-long coordinate.

Because the images are rectified, the MTT can assume a pure perspective projection using a pinhole camera model. The projection from SCC to image coordinates is:

$$\mathbf{P} = \mathbf{K}\left[\mathbf{R}_{\alpha\beta\gamma} \mid -\mathbf{p}_c\right] \tag{6.1}$$

where $\mathbf{p}_c$ is the position of the camera in SCC, $\mathbf{R}_{\alpha\beta\gamma}$ is a rotation matrix which rotates from camera coordinates to SCC and $\mathbf{K}$ is the intrinsic camera matrix

$$\mathbf{K} = \begin{pmatrix} f & s\cdot f & \frac{s_x-1}{2} \\ 0 & a\cdot f & \frac{s_y-1}{2} \\ 0 & 0 & 1 \end{pmatrix} \tag{6.2}$$

where f is the focal length

$$f = \frac{s_x}{2\tan\left(\frac{a_x}{2}\right)} \tag{6.3}$$

In order to project between the two coordinate systems, the MTT must therefore know certain parameters for each camera:

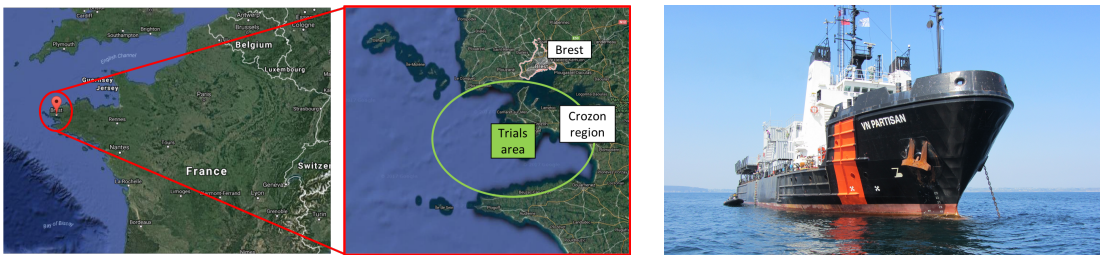
- Position in SCC: $\mathbf{p}_c = \begin{bmatrix} p_x & p_y & p_z \end{bmatrix}^T_{SCC}$
- Orientation: α (roll), β (pitch/tilt) and γ (yaw/pan)
- Image resolution: s_x and s_y in pixels
- Horizontal and vertical field of view: a_x and a_y in radians

These parameters were established for each camera during the set-up and calibration steps at each of the trials.

6.3 Real-world set-up and data collection

6.3.1 System deployment

Data collection and trials of the IPATCH system took place 3 times during the course of the project. In 2015 and 2016, a vessel called the *VN Partisan* was used to collect data and



(a) Region of the 2015 and 2016 IPATCH trials (b) *VN Partisan*

Fig. 6.5 Region and vessel used in the 2015 and 2016 campaigns

test the system in France. At the end of the project in 2017, the system was trialled on a tanker vessel called the *Kamari* in Greece. This section describes how the system was configured for each of the vessels.

VN Partisan

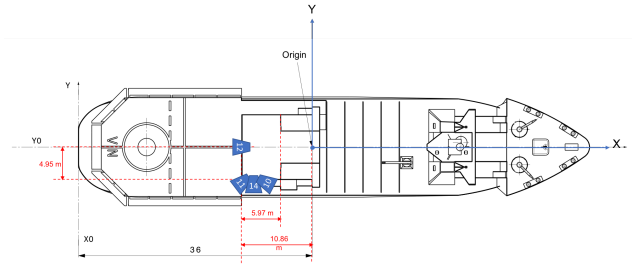
The 2015 and 2016 campaigns took place off the coast of France around the port of Brest and the Crozon region (Fig. 6.5a). The *VN Partisan* (Fig. 6.5b) is a multi-purpose offshore vessel used for training by the French Navy. It is approximately 79m long and 15m wide, and is conveniently equipped with training areas which include a fake bridge and cabins with space for the IPATCH system.

Cameras were installed on the roof of the training module with their fields of view covering approximately 180 degrees. In 2015, the cameras covered the area behind and starboard of the vessel; in 2016 the coverage was directly astern. These orientations were selected because most piracy attacks occur from behind or slightly to one side of the vessel. Fig. 6.6 shows the visual camera positions for 2015 and 2016 and a photo of the camera mountings.

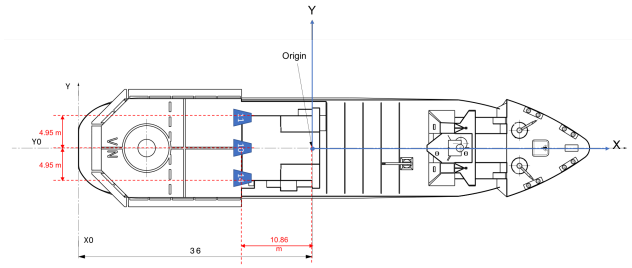
Two fast RHIBs were used to represent pirate skiffs, approximately 7m in length with maximum speeds of 25 and 50 knots. A fishing boat with maximum speed 8 knots was used to represent other non-pirate vessels, such as fishing boats and other transport craft. See Fig. 6.7 for pictures of the target vessels.

Kamari

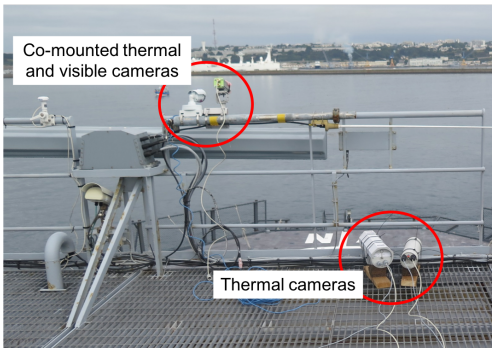
In 2017, a final demonstration of the system took place near Corinth in Greece on board the *Kamari* crude oil tanker. The *Kamari* is approximately 270m long and 45m wide and enabled the testing of the IPATCH system in a realistic context. Tankers are frequently



(a) 2015 campaign



(b) 2016 campaign



(c) Camera installation

Fig. 6.6 Camera installations on the *VN Partisan*



(a) *Bourre-pif*

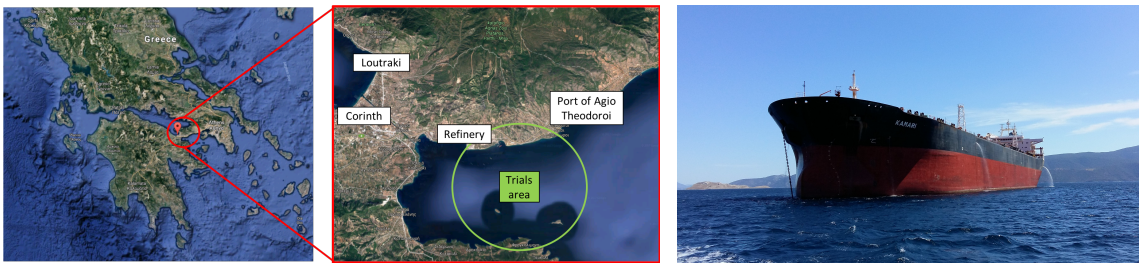


(b) *Black Bull*



(c) *Fishing Boat*

Fig. 6.7 Target skiffs from the 2015 and 2016 trials



(a) Region of the 2017 campaign (b) *Kamari*

Fig. 6.8 Region and vessel used in the 2017 campaign

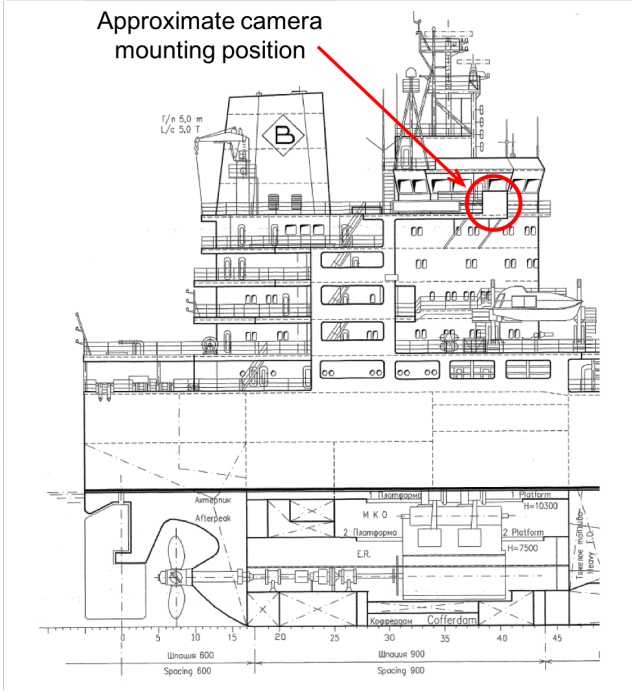
targeted by pirates and using the *Kamari* allowed the project to collect data with the representative distance scale and motion of a very large vessel. The *Kamari* transports oil around the world and often sails through high-risk piracy areas. Due to maritime regulations, the *Kamari* had to remain at anchor throughout the trials. However, attacks at anchorage are common in the real world, as this is when the vessel is most vulnerable.

One of the lessons learned from the 2015 and 2016 trials was the difficulty of attaching cameras to the vessel and reliably calibrating their positions and orientations. A special-purpose camera mounting frame was therefore designed and manufactured for the 2017 trials to hold four visual and three thermal cameras at fixed angles, so that appropriate field-of-view overlaps were guaranteed. An additional benefit of this was that the entire mounting frame could pre-assembled and calibrated before the trials and then easily attached to the ship on the day. Fig. 6.9 shows the mounting frame and its position on the Kamari. From this position, the cameras covered approximately 200 degrees to the stern and starboard side.

Three speed boats, similar to the size and speed of those used in 2015 and 2016, were used to represent pirate skiffs. These targets were different in appearance to those used previously. Whilst this was beneficial from the perspective of testing the algorithms with object appearances, the targets are all quite similar to each other. Fig. 6.10 shows pictures of the target skiffs from 2017.

6.3.2 Calibration and synchronisation

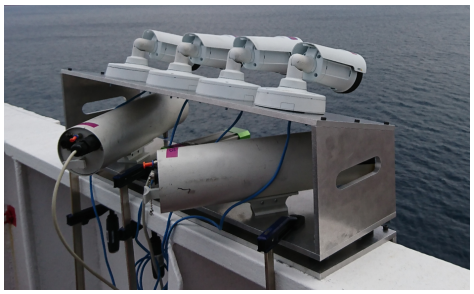
Good calibration and synchronisation of all the components of the system is important for reducing errors. However, accurate calibration and synchronisation on a large vessel at sea with limited time is difficult in practice. The intrinsic parameters of the cameras



(a) Camera installation location



(b) Camera installation on bridge wing



(c) Camera mounting frame

Fig. 6.9 Camera installation on the *Kamari*



(a) Skiff 1



(b) Skiff 2



(c) Skiff 3

Fig. 6.10 Target skiffs from the 2017 trials

were calibrated using a standard chessboard calibration procedure [213, 247]. This was done to produce a rectified image by reversing any spherical or tangential distortion components which are introduced by the camera lenses. The rectified images are used for object detection so that the MTT can assume a pure perspecitve projection model, as described in the previous section.

All the IPATCH system modules run on different computers. They are networked together in order to exchange information and one computer provided a time server with which all other computers could synchronise themselves. The cameras also connected to the same network, so their frame timestamps were also synchronised. However, the data from the vessel bridge systems and GPS tracking devices on the skiffs could only be synchronised approximately. Analysis after the trials showed that synchronisation was accurate to within 1 second for everything apart from the GPS trackers in the 2017 trials. An offset of approximately 5 seconds was found empirically by manual analysis of the data. This has been corrected for in the experiments in this chapter.

Measurement of the locations of the cameras on the vessel was approximate. Later testing with the data showed that that the projection of image detections to real-world tracks is not sensitive to camera height, other than at very low values (see also Section 4.4.3 and Fig. 4.27b). Given the positional accuracy of the projected positions, especially at long distances, the precise x and y location of the camera is also not important.

The most sensitive parameters were found to be the camera roll and pitch, and field of view angle. The on-site measurements of roll and pitch were crude and subsequently found to be a large source of error. Following the trials, the values were manually refined by replaying the data and aligning the projected image detections with the GPS groundtruth tracks. The situation was slightly better in 2017 due to the camera mounting frame.

6.4 Experiments

6.4.1 Implementation

During the IPATCH project, a multi-sensor multi-target tracking module was created [4]. A copy of the software was provided for use in this research. The system was run live during the trials in the project, but for these experiments, the system was run offline in the lab using recorded data. This provides more control over the experiments and allows to repeat things with different parameters. Also, the proposed object detection methods

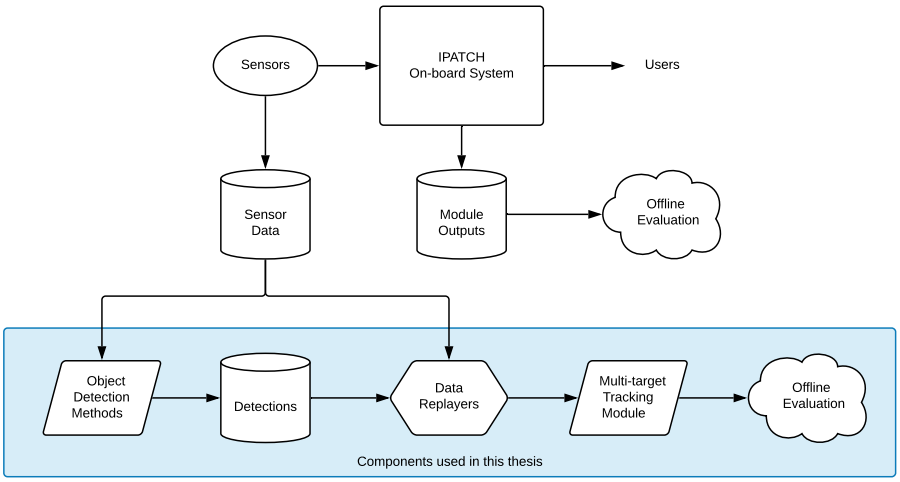


Fig. 6.11 Relationship of work in this thesis to the IPATCH system

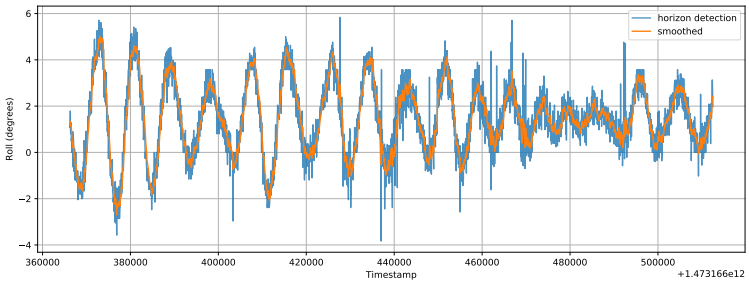
in this thesis were not complete during the live testing of the system. The relationship of the work in this thesis to the IPATCH system is depicted in Fig. 6.11.

The Multi-Target Tracker module (MTT) receives detections from the detection modules in the form of Protobuf[2] messages sent through the 0MQ messaging library[3]. In these experiments, the object detection was performed offline and the detections were stored. A script was written to play back the stored detections and other recorded sensor data (IMU, radar, etc.) to the MTT through the 0MQ Protobuf interface, as if they were running live (i.e. the timestamps were used to simulate real-time operation).

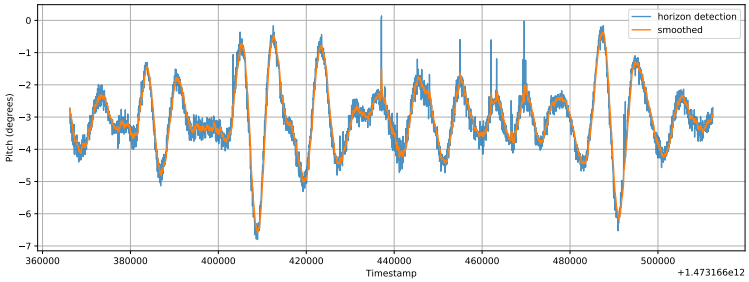
As described in Section 6.2.3, the MTT needs to know certain parameters for each camera in order to translate between image coordinates and ship-centred coordinates. In the implementation of the MTT, this is achieved by each object detection module sending the parameters inside its Protobuf messages. For the position of the camera ($\mathbf{p}_c$), field of view angles (a_x, a_y) and image resolution (s_x, s_y), the values are fixed and were read from a configuration file by the replayer script. The camera orientation values, however, depend on the motion of the vessel.

Originally, the navigational data provided by the bridge systems was intended to be used to update the camera orientation as the ship moves. However, the orientation of the camera with respect to the vessel was not measured sufficiently accurately. As explained in Section 6.3.2, the projection into SCC is unfortunately very sensitive to roll and pitch of

[2]https://developers.google.com/protocol-buffers
[3]http://zeromq.org



(a) Roll



(b) Pitch

Fig. 6.12 Effect of smoothing on roll and pitch estimation from horizon detection (Scenario 2016-Sc1_Tk5-CAM11)

the camera. Horizon detection was therefore used instead to estimate the instantaneous roll and pitch of the camera in each frame.

The horizon detection method presented in Chapter 4 was applied to all sequences. However, there is still some noise in the detected horizon lines. To reduce this, the horizon line values (y-position and rotation angle) were smoothed using the following expression with smoothing weight $\omega = 0.9$:

$$x'_t = \omega x'_{t-1} + (1-\omega)x_t \tag{6.4}$$

where x'_t is the smoothed value of x at timestep t. Fig. 6.12 shows some example results of this process.

6.4.2 Sequences

The experiments focus on three scenarios which represent common piracy attack patterns:

- In 2015-Sc3_Tk2, the skiffs are loitering at a distance from the vessel. They move slightly towards the vessel, but then suddenly approach the vessel at high speed.

- In 2016-Sc1_Tk5, a skiff stays close a fishing boat (trying to masquerade as a normal fishing boat). The skiff then peels off from the fishing boat and starts to follow / approach the vessel from the stern.
- In 2017-Sc3a, 3 skiffs appear from the distance and rapidly approach the vessel from different positions in a coordinated attack.

The sequences from cameras which capture all or most of the scenario action are selected to generate the object detections. These are also some of the sequences where image groundtruth is available and on which the proposed object detection methods were compared against the baselines in Chapters 4 and 5. This enables comparison between the image-based object detection performance and real-world tracking performance.

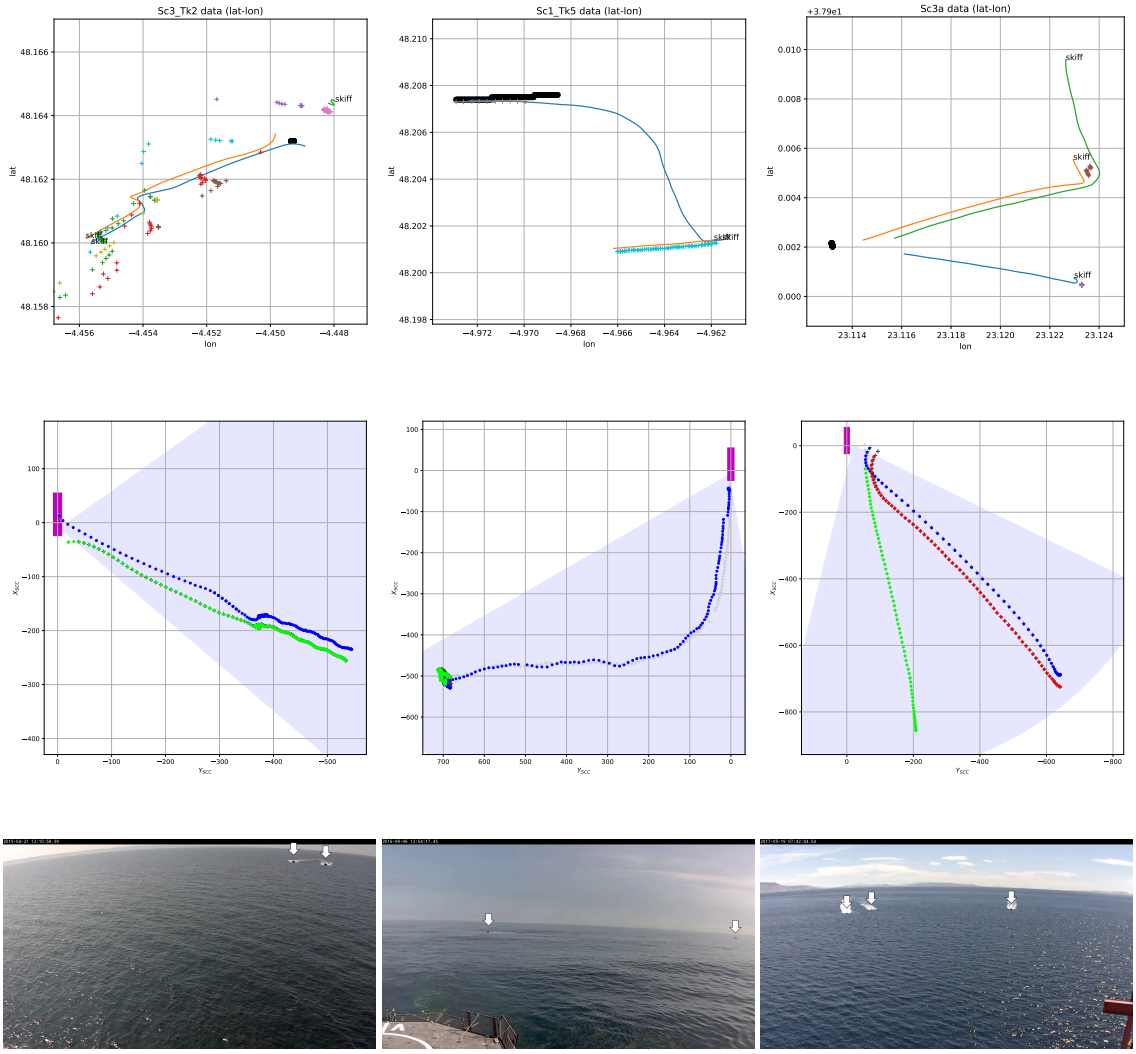
The data from the scenarios is plotted in Fig. 6.13 along with an example image from the camera sequence. In 2015-Sc3_Tk2 (Fig. 6.13a), the radar is only able to produce very noisy detections which are not well aligned with the skiffs. In 2016-Sc1_Tk5 (Fig. 6.13b), radar tracking very good for the larger fishing vessel (orange track) but only detects the skiff when it is very close (blue track). In 2017-Sc3a (Fig. 6.13c), the radar system was not able to acquire targets for the skiff by itself. Targets were manually acquired by a crew member on the bridge, but they were soon lost, hence they only appear as a small cluster of points near the start of the tracks. Fig. 6.14 shows the three skiffs appearing on the radar screen on-board the *Kamari*.

6.4.3 Evaluation procedure

The detections from the various modules are projected into the world coordinate system as points. The assumption is made that all objects will be on the surface of the sea, thus their z-coordinate is 0. The task under evaluation is therefore modelled as a 2D point-target tracking task. Extending the notation used in [85], the set of groundtruth tracks which exist at each timestep k is

$$\begin{aligned} \mathcal{G}(k) &= \{\mathcal{G}_1(k), \mathcal{G}_2(k), \ldots, \mathcal{G}_{M(k)}(k)\} \\ \mathcal{G}_m(k) &= \{t_k, \hat{\mathbf{x}}_k\} \end{aligned} \tag{6.5}$$

where $M(k)$ is the number of groundtruth tracks in timestep $k \in 1, 2, \ldots, K$, and $\mathbf{x}$ is the target state, represented by the position on the sea surface in ship-centered coordinates (SCC):



(a) 2015-Sc3_Tk2 (CAM14) (b) 2016-Sc1_Tk5 (CAM11) (c) 2017-Sc3a (CAM12)

Fig. 6.13 Plots of the three scenarios used to evaluate the object detection methods in a real-world context. Top row: tracks from radar (crosses), GPS groundtruth (lines) and the host vessel (black dots) in lat-long coordinates. Middle row: GPS groundtruth mapped to ship-centred coordinates (in metres), host vessel shown as pink rectangle to scale, field of view of camera shown as blue shaded region. Bottom row: frame from the sequence showing camera view, skiffs highlighted for clarity

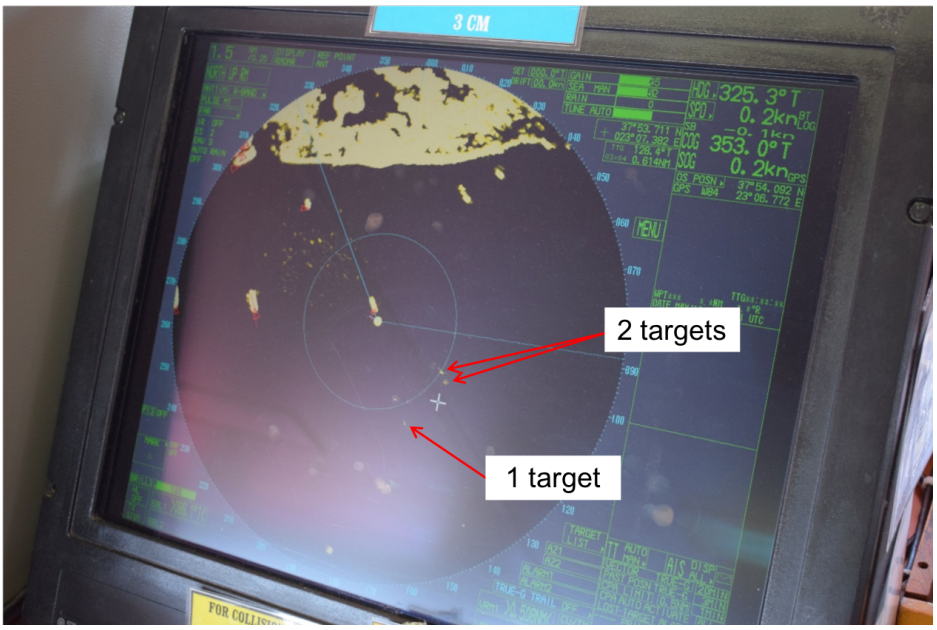


Fig. 6.14 Skiffs in 2017-Sc3a appearing on the radar screen on-board the *Kamari*. The radar system was not able to acquire the targets automatically so a crew member used the manual target acquisition function. However, the system was only able to track the skiffs for a short period of time.

$$\mathbf{x} = \begin{bmatrix} X \\ Y \end{bmatrix}_{SCC} \tag{6.6}$$

Similarly, the set of all tracks output by the MTT at each time*stamp* t is

$$\begin{aligned} \mathcal{T}(t) &= \{\mathcal{T}_1(t), \mathcal{T}_2(t), \ldots, \mathcal{T}_{N(t)}(t)\} \\ \mathcal{T}_n(t) &= \{t, \mathbf{x}_t\} \end{aligned} \tag{6.7}$$

and $N(t)$ is the number of tracks output by the MTT at each timestamp t. The time*stamp* of each time*step* is represented by t_k and is measured in milliseconds since the Epoch[4] in UTC[5]. The reason for having both time*steps* and time*stamps* is explained in the next section.

[4] 00:00:00 on 1st January, 1970
[5] Coordinated Universal Time timezone

Aligning MTT output with groundtruth

The groundtruth tracks of the target skiffs were captured using GPS devices which logged the positions every second, to the nearest second, producing a synchronous 1Hz stream. The MTT module estimates the number of targets and their states in an *asynchronous* manner in order to accommodate detections from multiple modules operating at different (and variable) frequencies. The timestamps in the MTT output are linked to the timestamps of the raw sensor data (e.g. each video frame) which were fused to produce them. The first challenge is therefore to match up the asynchronous MTT output with the groundtruth (GT) tracks.

First, the MTT and GT track points are filtered to excluded timestamps which lie outside the evaluation time range for the scenario and points which lie outside the camera's field of view. Then, each GT track is linearly interpolated for each timestamp in the set of timestamps from the MTT tracks. This creates a new set of GT tracks $\mathscr{G}'(t)$ in which each GT track $\mathscr{G}'_m(t)$ has a point for every timestamp in the MTT output:

$$\mathscr{G}'(t) = \{\mathscr{G}'_1(t), \mathscr{G}'_2(t), \ldots, \mathscr{G}'_{M(t)}(k)\} \tag{6.8}$$

where

$$\mathscr{G}'_m(t) = \{t, \hat{\mathbf{x}}_t\}, \quad \text{for } t \in \bigcup_{n=1}^{N(t)} \mathscr{T}_n(t) \tag{6.9}$$

The data is then grouped into timesteps by binning the GT and MTT points into scans of 1 second such that the centres of the bins are the timestamps from the original GPS groundtruth, t_k. The sets of groundtruth and MTT points that fall within timestep k are denoted

$$\mathscr{G}'_k = \{\mathscr{G}'(t)\} \text{ and } \mathscr{T}_k = \{\mathscr{T}(t)\}, \quad \text{where } t_k - 500\,\text{ms} \leq t < t_k + 500\,\text{ms}. \tag{6.10}$$

Some timesteps will be empty if there are no groundtruth targets visible in the scene and no MTT track points. False positive detections can be identified by timesteps with MTT track points but no groundtruth targets, and missed detections are the reverse. Timesteps which contain both MTT and groundtruth track points can be evaluated for localisation accuracy and other metrics, and the above process guarantees that there is a groundtruth point in every track for each MTT track point. Fig. 6.15 illustrates the timestep binning concept.

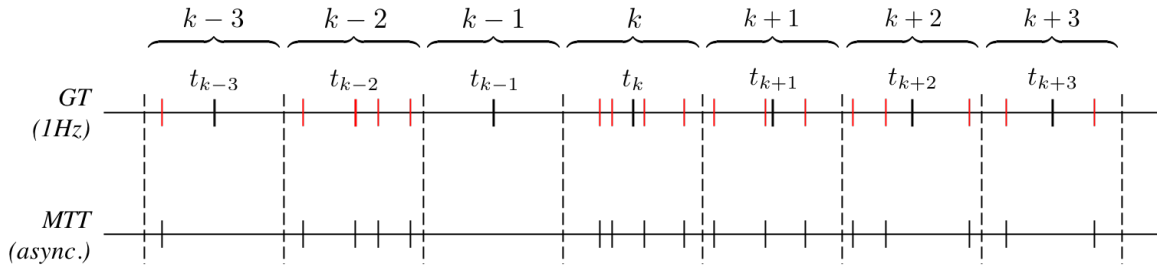


Fig. 6.15 Illustration of the timestep binning and alignment concept. The GT track is initially a synchronous 1Hz stream and is linearly interpolated for the timestamps from the MTT output (red tickmarks). The track points are binned into timesteps which enable alignment and evaluation between the synchronous groundtruth and asynchronous MTT streams.

Point set assignment is performed for each timestamp within a timestep using the Hungarian algorithm [124, 150] and the Euclidean distance between points as the cost function. Gating is used such that points are unassigned if their distance exceeds a threshold, d_g. Due to the high sensitivity of position estimation caused by the projection, a value of 100m is used for d_g.

A record is kept of how many times each MTT track was assigned to each GT track within the timestep. A second Hungarian optimisation is then performed to determine which MTT track is assigned to which GT track (if any) for that timestep. In most cases, this equates to the MTT track with the highest number of assignments for that GT track, but the Hungarian assignment is needed to resolve edge cases where tracks could be assigned twice.

Based on the track-to-track assignment, the number of true positives (TP), missed tracks (MT) and false tracks (FT) can be determined for each timestep. The IDs of the MTT tracks which were assigned to each GT track are also logged for analysing ID changes. The results of this process are then used to compute the performance evaluation metrics.

Metrics

The object detection and tracking stage needs to provide high-quality tracks to the later stages of the pipeline (situational awareness, threat recognition). Properties of a tracker which are considered important by operators of military systems include accuracy, completeness, continuity, ambiguity and timeliness [21, 22, 85, 186]. Many metrics can be

defined under these categories. In these experiments, the focus is on performance aspects which are important in the piracy detection task:

- **Positional accuracy** – how closely does the estimated position of the tracked targets match the true positions. Good position accuracy is important for estimating how far away the pirates are and what speed they are travelling, and therefore how quickly they will intercept the vessel. Knowing the location is also important for situational awareness of other non-threatening vessels and in other applications, such as collision avoidance.
- **Tracking completeness / missed tracks** – what proportion of the target's trajectory is tracked. This is important, as location cannot be estimated while the target is not being tracked.
- **False tracks** – in the piracy context, it is clearly important not to miss any targets, but false alarms distract the crew on the bridge and a high false alarm rate could lead to the system being mistrusted or ignored completely.
- **Continuity / fragmentation** – how often the track changes ID (i.e. appears as a new target). This is important for higher levels of processing, which may not work properly if they think new targets are appearing/disappearing or there is a sudden change of course due to an ID swap.
- **Detection distance / time to detect** – how quickly a target is acquired by the tracking system and at what distance. This is key to achieving good early warning capabilities to give the crew as much time as possible to respond to a potential threat. There is a strong dependence on the detection module capabilities, as the tracker cannot know about targets which are not detected, but detections must also be of sufficient quality for the tracker to initialise and output a track.

True Positives, Missed Tracks and False Tracks After MTT tracks have been assigned to GT tracks, the number of correctly tracked targets (True Positives, TP) in timestep k is:

$$TP(k) = |\mathscr{S}_k| \tag{6.11}$$

where $\mathscr{S}_k = \{(\mathscr{G}'(t), \mathscr{T}(t))\}$ is the set of matched tracks for timestep k and $|.|$ is the cardinality operator. Any GT tracks and MTT tracks which did not get assigned are classified as

Missed Tracks (MT) and False Tracks (FT), respectively:

$$MT(k) = |\{\mathscr{G}'(t) \in \mathscr{G}'_k \mid \mathscr{G}'(t) \in \mathscr{S}_k\}| \tag{6.12}$$

$$FT(k) = |\{\mathscr{T}(t) \in \mathscr{T}_k \mid \mathscr{T}(t) \notin \mathscr{S}_k\}| \tag{6.13}$$

For sequence-level evaluation, the mean number of false tracks is computed across all timesteps:

$$\widehat{FT} = \frac{1}{K}\sum_{k=1}^{K} FT(k) \tag{6.14}$$

Completeness and Precision Track completeness is the proportion of the track which was correctly tracked, i.e. the number of times a target was tracked divided by the total track length up to timestep k. The Completeness score (Cp) is calculated across all tracks in the groundtruth. This is equivalent to the True Positive Rate. Precision (Pr) measures the proportion of tracks output by the MTT that were matched to groundtruth tracks.

$$Cp(k) = \frac{\sum_{k'=1}^{k} TP(k')}{\sum_{k'=1}^{k} M(k')} \tag{6.15}$$

$$Pr(k) = \frac{\sum_{k'=1}^{k} TP(k')}{\sum_{k'=1}^{k} N(k')} \tag{6.16}$$

Positional error Absolute positional error p_{abs} is calculated using the Euclidean distance between points. The mean error is calculated for all matched tracks at each timestamp t and over all timestamps in the timestep, $\mathbf{t}_k$:

$$p_{abs}(k) = \frac{1}{|\mathbf{t}_k|}\sum_{t\in\mathbf{t}_k}\left(\frac{1}{|\mathscr{S}_t|}\sum_{i\in\mathscr{S}_t}||\mathbf{x}_t^{(i)} - \hat{\mathbf{x}}_t^{(i)}||\right) \tag{6.17}$$

Because of the large errors introduced by the camera calibration and other factors described in the previous section, the relative error p_{rel} is also computed:

$$p_{rel}(k) = \frac{1}{|\mathbf{t}_k|}\sum_{t\in\mathbf{t}_k}\left(\frac{1}{|\mathscr{S}_t|}\sum_{i\in\mathscr{S}_t}\frac{||\mathbf{x}_t^{(i)} - \hat{\mathbf{x}}_t^{(i)}||}{||\hat{\mathbf{x}}_t^{(i)}||}\right) \tag{6.18}$$

To give a sequence-level score, absolute and relative error are averaged across all timesteps, assigning $p_{abs} = d_g$ and $p_{rel} = 1$ for timesteps where no assignments were made:

$$\widehat{p_{abs}} = \frac{1}{K} \sum_{k=1}^{K} p_{abs}(k) \tag{6.19}$$

$$\widehat{p_{rel}} = \frac{1}{K} \sum_{k=1}^{K} p_{rel}(k) \tag{6.20}$$

Continuity The continuity or fragmentation of the tracks can be measured by counting the number of times the assigned MTT track ID changes. A groundtruth track has low continuity / high fragmentation if it is frequently assigned MTT tracks with different IDs. Although it was designed for image-based tracking evaluation, Normalised ID changes (NIDC) [151] is adopted because it normalises for different track lengths. The NIDC score for groundtruth track i at timestep k is calculated as

$$\mathrm{NIDC}_i(k) = \frac{\mathrm{IDC}_i(k)}{\mathrm{IDC}_i^{max}(k)} \tag{6.21}$$

where $\mathrm{IDC}_i^{max}(k)$ is the maximum possible number of ID changes for the track at timestep k:

$$\mathrm{IDC}_i^{max}(k) = \left|\mathscr{G}_i'(k)\right| - 1 \tag{6.22}$$

The NIDC score is computed as an average over all groundtruth tracks:

$$\mathrm{NIDC}(k) = \frac{1}{M(k)} \sum_{i=1}^{M(k)} \mathrm{NIDC}_i(k) \tag{6.23}$$

6.4.4 Sources of error

As discussed in previous sections, there are some sources of significant error which are independent of the object detection methods under evaluation:

- Calibration of camera intrinsics and distortion parameters
- Calibration of camera focal length / horizontal field of view
- Camera roll and tilt estimated from the horizon line

In addition, discrete image coordinates (pixels) create an error which gets worse with distance. Even with perfect camera calibration and pose estimation, location estimation is extremely sensitive to small changes in image position. A difference of 1 pixel can correspond to a 40m error at 1km, and 462m at 4km.

For this reason, two additional 'detection' sources are processed through the MTT: the GPS groundtruth tracks (which are ingested as if they were radar detections) and the image groundtruth annotations (which are rectified and then ingested as bounding boxes in the same way as the object detection methods under analysis). The idea is to assess the scale of the "system error" (i.e. how much the camera calibration, horizon detection and tracking process contributes to positional error and other performance metrics) to put the results from the object detection methods in context.

6.4.5 Summary of MTT inputs

Object detections from a number of different sources are fed into the MTT to compare performance. The GPS and bounding box groundtruth inputs as described above will show how the system responds under ideal conditions (perfect detections). The proposed object detection methods from Chapters 4 and 5 and three baseline methods as described in Chapter 3 (TSFC [156], Mask R-CNN [89] and YOLO [174]) will be compared to see how the characteristics of their output impacts on tracking performance. Note that IMBS is not included as the number of false positive detections was so high, the MTT was overloaded. Finally, input from radar and thermal cameras recorded in the trials is used to show the comparison with the visual detection methods. The inputs are summarised in Table 6.1.

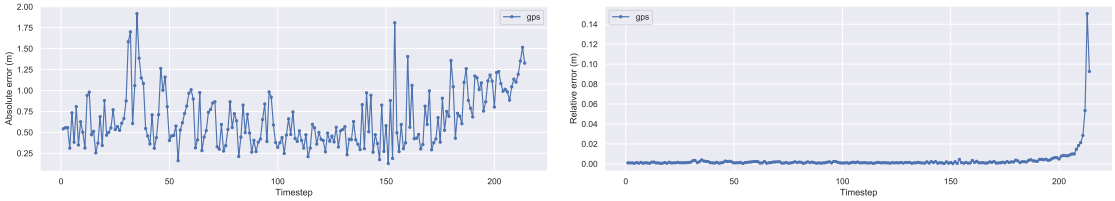
Table 6.1 Summary of detection sources processed by the MTT

Name	Description
gps	Groundtruth tracks from GPS trackers
boxes	Image groundtruth bounding boxes
rad	Recorded radar data from the trials
thermal	Recorded object detections from thermal cameras from the trials using the method in [18]
maskrcnn	Mask R-CNN [89] R-50 variant
YOLO	YOLO v3 variant [174]
TSFC	Temporally stable feature clusters [156]
saliency	Proposed method from Chapter 4 (saliency-995-99-d8-hv-depth)
semantic	Proposed method from Chapter 5 (EDANet_boundary-coordconv)

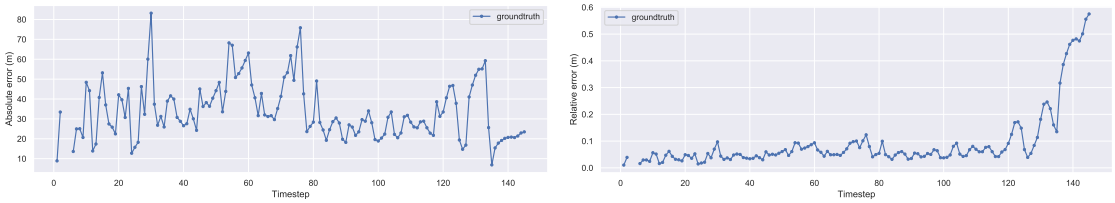
6.5 Results and analysis

6.5.1 System error

Fig. 6.16 shows examples of absolute and relative positional errors for two sequences when the GPS and image groundtruth is fed into the MTT. The MTT tracking process introduces less than 5m of positional error. As the skiffs get closer, the relative error increases but is still small. Much larger errors (up to 80m) are observed when the image groundtruth is used. These are caused by discrete pixel locations of the bounding boxes and variation in their placement by the human annotator. These small errors are multiplied in the projection process by errors in the roll and pitch estimation caused by imperfect horizon detection. The relative error reflects the fact that a small difference in bounding box position creates larger errors at larger differences. The large errors show that a gating distance of 100m was not unreasonable.



(a) GPS, 2015-Sc3_Tk2



(b) boxes, 2016-Sc1_Tk5

Fig. 6.16 Absolute (left) and relative (right) errors for GPS and bounding box inputs

6.5.2 Performance using object detection methods

Tables 6.2a, 6.2b and 6.2c show the sequence level results for the visual object detection inputs in terms of absolute and relative error, tracking completeness and precision, mean false positives and normalised ID changes. Figs. 6.17, 6.18 and 6.19 display the actual tracks produced by the MTT based on the inputs from the object detection methods.

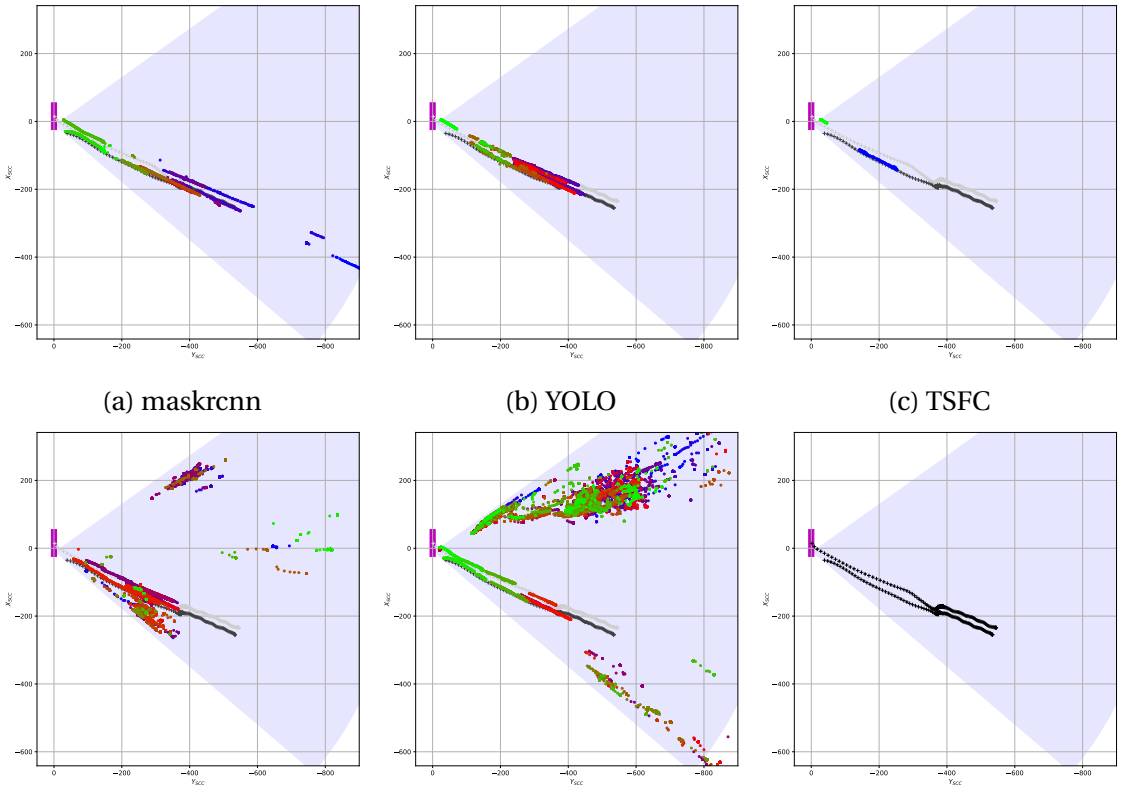
Positional error is highest overall in 2015-Sc3_Tk2, even though the tracks look well-aligned visually (Fig. 6.17). This is also the case for the bounding box input, which suggests that roll and pitch estimation using horizon detection is a significant source of error. The saliency and semantic methods detect glare and reflections to the left of the field of view which are persistent enough to be tracked by the MTT, leading to higher numbers of false tracks. This can also be seen visually in Fig. 6.17. Nevertheless, saliency achieves the best positional accuracy.

An unexpected result is the perfect score achieved by TSFC for precision, false tracks and NIDC. Inspection of the tracking output in Fig. 6.17 reveals that very little of the target trajectories was actually tracked. The scores appear artificially high because of the low number of detections overall (for example, if there are no detections, there cannot be any false positives either). The situation is the same in the 2016-Sc1_Tk5 sequence.

The 2016-Sc1_Tk5 scenario was the most challenging for all methods, resulting in lower precision, and higher false tracks and ID changes. The motion of the vessel in this sequence is significant which leads to large variation of position for distant targets. This can be seen in Fig. 6.18. Despite the chaotic appearance of the plots, the methods achieve relatively good positional errors on average over the sequence, with the semantic approach out-performing what was achieved using the bounding boxes (although this is probably more a reflection of the difficulty in placing bounding boxes for this sequence).

In 2017-Sc3a, the maskrcnn, YOLO and saliency inputs produce good positional accuracy and precision again. Tracking completeness is particularly good for saliency and YOLO. As with the other sequences, the saliency and semantic inputs produce a lot of false positives, however their tracks are more continuous (lower NIDC) than the other methods. Compared to the other sequences, the TSFC method is able to find more stable features resulting in more track output. Its false positive and precision scores are still low (due to the number of tracks still being small) but there is more fragmentation in the tracking, indicated by a higher NIDC score.

As described previously, the critical performance criterion for piracy early warning is how quickly the detection and tracking system can identify an object. The time until the first detection was measured for each method by recording the timestep of the first true positive track point. Table 6.3 shows the results. The proposed saliency method was the first method to output a detection for the 2017 sequence, and joint first with the semantic



(a) maskrcnn (b) YOLO (c) TSFC

(d) saliency (e) semantic (f) Groundtruth tracks

Fig. 6.17 MTT outputs for scenario 2015-Sc3_Tk2. Grey tracks are GPS groundtruth, coloured tracks are MTT output.

method for the 2016 sequence. However, over all the sequences, maskrcnn and YOLO consistently detected the targets early on.

6.5.3 Comparison with radar

Fig. 6.20 shows the tracks produced by the MTT when radar is used. Visually, it can be seen that the radar tracks do not align well with the groundtruth and most of the tracks are not detected. In 2015 and 2016, the radar used was a more advanced system with built-in tracking capabilities. In 2016-Sc1_Tk5 (Fig. 6.20b), the stationary target (a fishing boat) is tracked quite well (blue track) but the smaller, moving skiff is only tracked when it gets close to the vessel (green track). In 2015, neither skiff is detected consistently and there is a lot of noise from the environment. In 2017, the radar was a much more basic navigational radar. As described in Section 6.4.2, the radar system was not able to acquire targets for the skiff by itself. Targets were manually acquired by a crew member on the

Table 6.2 Sequence-level results for the visual object detection methods: mean absolute positional error ($\widehat{p_{abs}}$), mean relative positional error ($\widehat{p_{rel}}$), completeness (Cp), precision (Pr), mean false tracks ($\widehat{FT}$) and NIDC. For positional errors, mean false tracks and NIDC, lower is better. Best and second best highlighted in green and blue, respectively (boxes not included when computing best).

(a) 2015-Sc3_Tk2

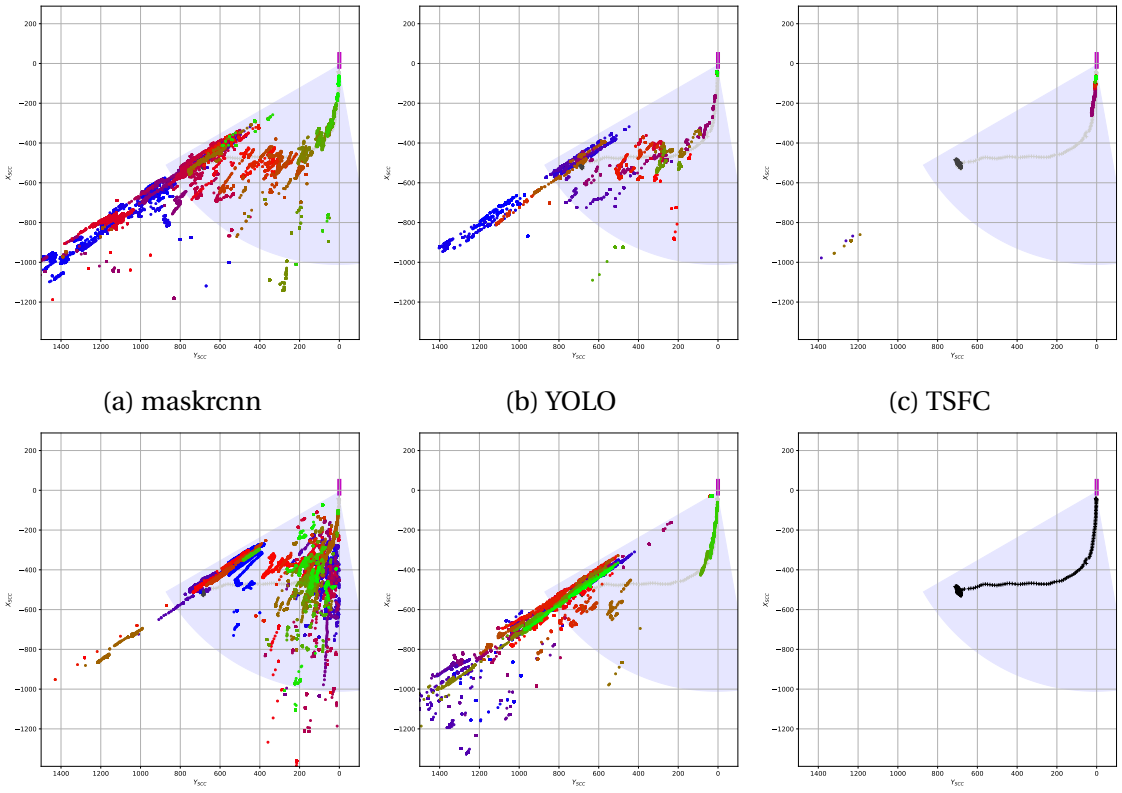
Method	$\widehat{p_{abs}}$	$\widehat{p_{rel}}$	Cp	Pr	$\widehat{FT}$	$NIDC$
boxes	56.3	0.411	0.668	0.667	0.7	0.030
maskrcnn	68.1	0.679	0.220	**0.531**	**0.4**	**0.042**
YOLO	**66.8**	**0.570**	**0.441**	0.423	1.2	0.122
TSFC	94.3	0.959	0.033	**1.000**	**0.0**	**0.000**
saliency	**64.3**	**0.452**	**0.456**	0.173	4.3	0.070
semantic	75.0	0.733	0.208	0.027	15.1	0.103

(b) 2016-Sc1_Tk5

Method	$\widehat{p_{abs}}$	$\widehat{p_{rel}}$	Cp	Pr	$\widehat{FT}$	$NIDC$
boxes	35.6	0.117	0.943	0.198	7.4	0.113
maskrcnn	**38.5**	**0.132**	**0.862**	0.130	11.1	0.277
YOLO	51.6	0.307	0.654	0.132	**8.3**	**0.112**
TSFC	78.0	0.794	0.127	**0.278**	**0.7**	**0.041**
saliency	39.7	0.236	**0.709**	**0.138**	8.6	0.193
semantic	**35.5**	**0.173**	0.704	0.090	13.6	0.221

(c) 2017-Sc3a

Method	$\widehat{p_{abs}}$	$\widehat{p_{rel}}$	Cp	Pr	$\widehat{FT}$	$NIDC$
boxes	23.9	0.078	1.000	0.925	0.2	0.005
maskrcnn	**29.0**	**0.156**	**0.841**	**0.535**	**2.0**	0.151
YOLO	**31.5**	**0.177**	**0.839**	0.415	3.3	0.121
TSFC	39.4	0.357	0.488	**0.527**	**1.2**	0.165
saliency	32.1	0.197	0.534	0.240	4.8	**0.099**
semantic	45.1	0.425	0.437	0.211	4.6	**0.092**



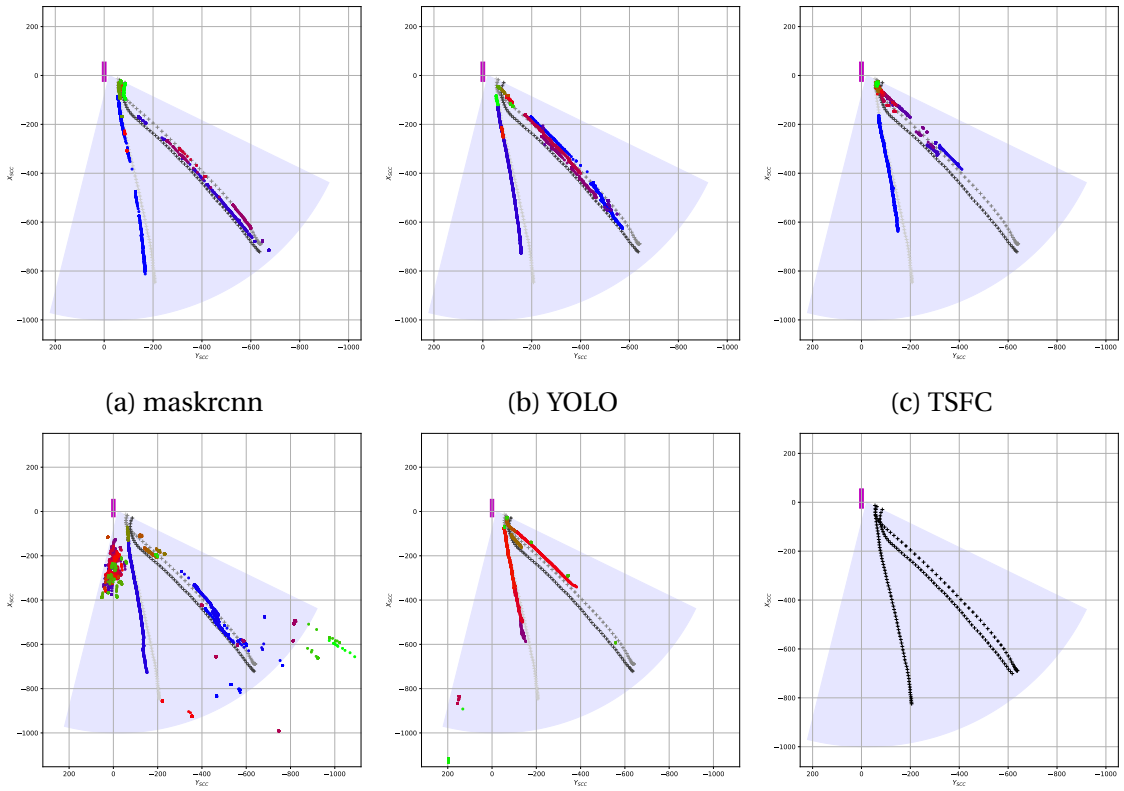
(a) maskrcnn (b) YOLO (c) TSFC

(d) saliency (e) semantic (f) Groundtruth tracks

Fig. 6.18 MTT outputs for scenario 2016-Sc1_Tk5. Grey tracks are GPS groundtruth, coloured tracks are MTT output.

Table 6.3 Time to detection, measured as the first timestep with a True Positive track detection.

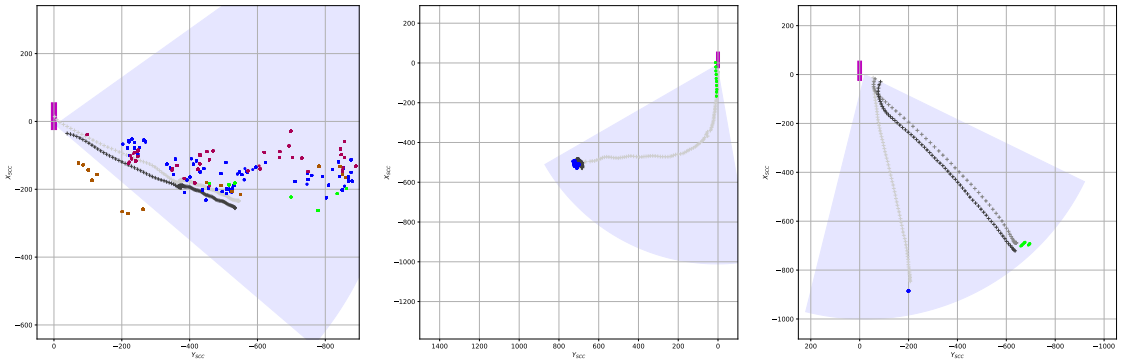
Sequence	maskrcnn	YOLO	TSFC	saliency	semantic
2015-Sc3_Tk2-CAM14	16	17	192	40	72
2016-Sc1_Tk5-CAM11	3	11	110	1	1
2017-Sc3a-CAM12	3	6	21	2	27



(a) maskrcnn (b) YOLO (c) TSFC

(d) saliency (e) semantic (f) Groundtruth tracks

Fig. 6.19 MTT outputs for scenario 2017-Sc3a. Grey tracks are GPS groundtruth, coloured tracks are MTT output.



(a) 2015-Sc3_Tk2 (b) 2016-Sc1_Tk5 (c) 2017-Sc3a

Fig. 6.20 Tracks using the radar detections as input to the MTT. Grey tracks are GPS groundtruth, coloured tracks are MTT output.

bridge at the start of the scenario (see Fig. 6.14) but Fig. 6.20c shows that they were not tracked for long.

Table 6.4 compares the quantitative results from the visual camera object detection inputs with results from radar input. Despite poor visual performance, radar scores well in terms of false tracks, precision and NIDC. This is due to the very low number of detections overall. In 2016-Sc1_Tk5, the positional accuracy is good (low error) due to the fishing vessel being tracked well. However, track completeness is not as good, as the other skiff is not tracked at all until near the end of the sequence. Completeness and accuracy are both poor in 2017-Sc3a which reflects the situation discussed above.

Table 6.4 Sequence-level results comparing radar against the visual object detection methods: mean absolute positional error ($\widehat{p_{abs}}$), mean relative positional error ($\widehat{p_{rel}}$), completeness (Cp), precision (Pr), mean false tracks ($\widehat{FT}$) and NIDC. For positional errors, mean false tracks and NIDC, lower is better. Best and second best highlighted in green and blue, respectively.

(a) 2015-Sc3_Tk2

Method	$\widehat{p_{abs}}$	$\widehat{p_{rel}}$	Cp	Pr	$\widehat{FT}$	$NIDC$
rad	**65.8**	**0.563**	0.343	0.211	2.6	**0.059**
maskrcnn	68.1	0.679	0.220	**0.531**	**0.4**	**0.042**
YOLO	66.8	0.570	**0.441**	**0.423**	**1.2**	0.122
saliency	**64.3**	**0.452**	**0.456**	0.173	4.3	0.070
semantic	75.0	0.733	0.208	0.027	15.1	0.103

(b) 2016-Sc1_Tk2

Method	$\widehat{p_{abs}}$	$\widehat{p_{rel}}$	Cp	Pr	$\widehat{FT}$	$NIDC$
rad	**14.1**	**0.064**	0.581	**0.942**	**0.1**	**0.003**
maskrcnn	38.5	**0.132**	**0.862**	0.130	11.1	0.277
YOLO	51.6	0.307	0.654	0.132	**8.3**	**0.112**
saliency	39.7	0.236	**0.709**	**0.138**	8.6	0.193
semantic	**35.5**	0.173	0.704	0.090	13.6	0.221

(c) 2017-Sc3a

Method	$\widehat{p_{abs}}$	$\widehat{p_{rel}}$	Cp	Pr	$\widehat{FT}$	$NIDC$
rad	81.0	0.750	0.167	**1.000**	**0.0**	**0.000**
maskrcnn	**29.0**	**0.156**	**0.841**	**0.535**	**2.0**	0.151
YOLO	**31.5**	**0.177**	**0.839**	0.415	3.3	0.121
saliency	32.1	0.197	0.534	0.240	4.8	0.099
semantic	45.1	0.425	0.437	0.211	4.6	**0.092**

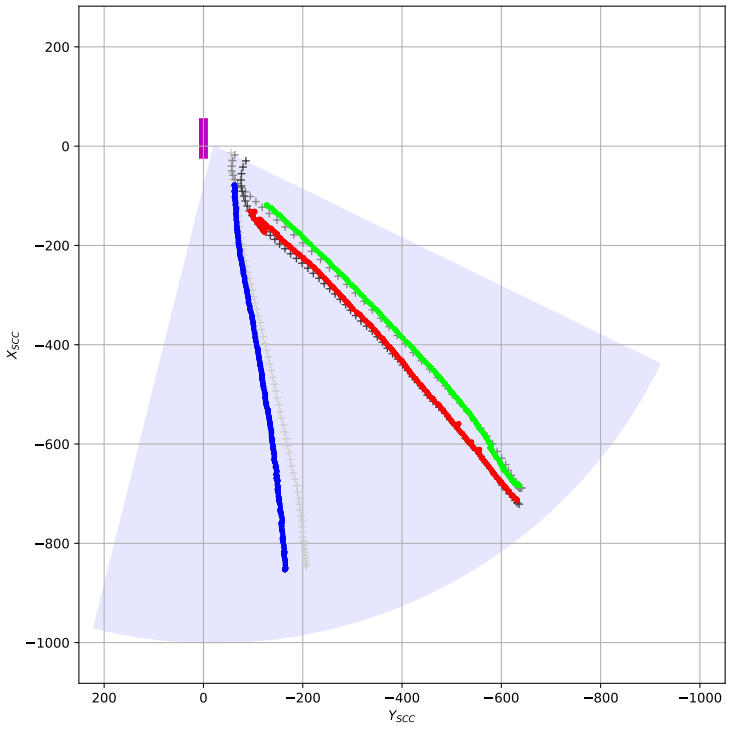


Fig. 6.21 Tracks from the thermal cameras for 2017-Sc3a

6.5.4 Comparison with thermal cameras

Detections from the thermal cameras were only available for one sequence but it is clear that their detection performance is superior to any of the visual camera methods (see Table 6.5 and Fig. 6.21). They are also superior to the results achieved using the image bounding box annotations. This highlights the issues with human error in bounding box placement, as discussed in Chapter 3.

The thermal detection method used is based on simple background subtraction plus a more sophisticated tracking process in image space [18]. The likely reasons for the high performance are:

- High contrast between the targets and sea, and reduced clutter from dynamic background in the thermal spectrum
- Tracking in image space before outputting detections
- Smaller field of view angle of the thermal cameras

The first factor means that background subtraction is an effective detection method with high localisation accuracy and minimal false positives. The second factor reliably

Table 6.5 Comparison of visual object detection methods with object detections from thermal cameras

Method	$\widehat{p_{abs}}$	$\widehat{p_{rel}}$	Cp	Pr	$\widehat{FT}$	$NIDC$
boxes	**23.9**	**0.078**	**1.000**	**0.925**	**0.2**	**0.005**
maskrcnn	29.0	0.156	0.841	0.535	2.0	0.151
YOLO	31.5	0.177	0.839	0.415	3.3	0.121
TSFC	39.4	0.357	0.488	0.527	1.2	0.165
saliency	32.1	0.197	0.534	0.240	4.8	0.099
semantic	45.1	0.425	0.437	0.211	4.6	0.092
thermal	**14.1**	**0.031**	**0.904**	**1.000**	**0.0**	**0.000**

fills any missed detections, improves the estimation of position over time, and removes most (if not all) false positives. The third factor relates to size of the targets in the image. The horizontal field of view (FoV) determines the pixel footprint in the real world. The thermal cameras used in the IPATCH trials had a FoV of 25 degrees, which corresponds to a pixel footprint of 1m at a distance of 2.9km. The visual cameras had FoV values up to 100 degrees, which corresponds to 1.4km for a 1m footprint. At a given distance, a target will therefore appear larger (more pixels) in the thermal cameras than in the visual ones.

6.6 Summary

This chapter investigated the performance of six object detection methods in the context of a complete maritime surveillance system using data collected in the field as part of the IPATCH project. The objective was to analyse how different characteristics of object detection methods affect how useful their detections are to the later stages of the surveillance pipeline (tracking, situational awareness, threat recognition, etc.).

The on-board surveillance system was described to explain how detections from the object detection methods get processed by other modules of the system. The Multi-Target Tracking (MTT) module was presented in more detail, as this performs the important process of converting image-based detections into tracks in the real world. The sources of error in this process were discussed, as they play an important role in the performance analysis of the tracking output. The real-world set-up of the two trials in France and Greece were described to explain how data was collected on-board real vessels at sea. The offline experimental set-up was explained, along with the performance evaluation procedure and metrics. The objective was to give an assessment of performance which is relevant to the end users of a maritime piracy surveillance system.

In general, results were consistent with those from previous chapters, namely that the deep learning-based methods (Mask R-CNN R50 and YOLO v3) perform well 'out of the box'. When detections are made, they are typically well-localised on the true target, and the number of false positives are very low. The exception to this is very small, distant targets which are missed by Mask R-CNN and YOLO, but can be detected by the proposed saliency and semantic segmentation-based methods. The saliency and semantic segmentation-based methods generate a lot of false tracks (caused by reflections, glare and other distractors which are not transient enough to be filtered out). Position errors for these methods are not as good as the Mask R-CNN and YOLO values, but they are still within reasonable ranges for detection distances of 100s of metres. Track continuity is often better, due to the larger number of detections overall.

In the final chapter, conclusions from all chapters are consolidated and the main contributions of this research are summarised. Future work is also proposed to extend the experiments and analysis.

Chapter 7

Conclusions and Future Work

7.1 Findings and limitations

The empirical analysis has shown that the proposed methods are feasible, but further work is required to improve their performance. It is also likely that additional effort spent on optimising the implementation could bring their processing speeds closer to the real-time goal for an operational system.

7.1.1 Saliency-based object detection

The saliency method relied on the assumption that objects would be salient compared to the background. Under these conditions, detection was quite reliable, even with very small distant targets, although the spreading effect of the saliency map post-processing step often caused bounding boxes to be overestimated, or merged if two objects were close together. In the open sea, the assumption will hold in many cases. In the piracy attack scenario, for example, there will only be a few skiffs and the background will be open sea and sky. However, the assumption breaks down in a number of cases:

- The presence of land in the background creates a salient region which may either be detected as a false positive, or increase the overall level of saliency, thereby making the true targets less salient by comparison (under the percentile thresholding mechanism).
- Scenes containing many objects create a similar problem, as objects are forced to compete with each other. This is similar to what happens in the human visual

system. There, higher-level processing takes over to either suppress some regions or visit regions in turn.

- Wake and bright reflections from the water are, by definition, salient. Whilst the horizontal-vertical thresholding and scene depth map weighting steps mitigated this to a certain extent, there is still work to be done in eliminating these false detections completely.

7.1.2 Semantic segmentation-based object detection

The idea behind the semantic segmentation-based approach was to invert the object detection problem and instead analyse the main regions of the scene. In the maritime case, this only consists of a few classes, as the majority of the background is either sea or sky. Object detection is framed as the problem of finding things which do not look like one of these two classes. Even with very limited training data from a different domain, EDANet [139] was able to detect objects through this method, including some quite challenging targets. The main observations were:

- A lot of false positive detections were created. The relatively simple rule-based reasoning can filter out some obvious cases, but more sophisticated reasoning or some secondary process would be needed to tell the difference between real objects and 'regions which are not classified as sea or sky'.

- A fundamental limitation of semantic segmentation is the inability to distinguish between very close or overlapping objects. This has been known for a long time and *instance* segmentation is now an active area of research which could be explored to resolve this (Mask R-CNN [89] is an example).

- CoordConv and multi-task learning had very inconsistent effects across the datasets and sequences, despite seeming very promising at first. The results indicate that boundary prediction improves performance, which reflects what is reported elsewhere in the literature, but further experiments would be required using richer training and evaluation data.

7.1.3 Visual object detection in the real world

As is the case across a wide range of applications nowadays, the deep learning approaches proved to be superior. However, their performance in maritime surveillance is noticeably less than that in more classical object detection domains. There is scope for further research to adapt deep neural networks for maritime surveillance. At the same time, there is still a role for more classical methods, as shown with the detection of very small and distant objects. Ideally, multiple approaches should be used together in a system to balance their strengths and weaknesses.

The trade-off between missed detections and false positives is still a challenge. The background subtraction approach (IMBS [30]) revealed an extreme case where the rest of the surveillance system was overloaded by the number of false detections. In contrast, the TSFC [156] approach produced very few false positives but struggled to detect objects consistently enough for them to be tracked at all.

The main limitation of the system as a whole was the estimation of the camera pose (roll and pitch). Whether done through online horizon detection or by using data from the ship's IMU sensor, small errors in angle create very large positional errors. The system is also very sensitive to the bounding box positions of points near the horizon: a difference of 1 pixel can correspond to a 40m error at 1km, and 462m at 4km.

Finally, this work has highlighted the importance of evaluating academic methods in realistic scenarios and under real-world conditions, as well as on carefully curated benchmark datasets. This would drive development of algorithms and implementations which can be more readily deployed in real-life applications or developed into products.

7.2 Outcomes against objectives

The main objective of this research was to develop object detection methods which had the following important properties for use in maritime surveillance and piracy detection:

- Can detect small, fast-moving skiffs approaching the vessel as early as possible to maximise warning for the crew.
- Provide high quality detections to the higher-level stages of an on-board piracy surveillance system to support tracking, situational awareness and threat detection.

- Do not make strong assumptions about the appearance of the target or scene so that they can be used in a wide range of contexts and applications.
- Are robust to camera motion, wake, reflections and environmental conditions.
- Can operate in real-time.

Whilst further evaluation should be carried out, the results showed that the proposed saliency and semantic segmentation methods can detect small targets and exhibit performance which improves on previous work. The more recent deep learning-based object detectors were generally superior, but the proposed methods provide sufficient detection accuracy and consistency to support tracking in the surveillance system.

The proposed methods make very few assumptions about the scene and potential objects. This means they can, in principle, detect any object which appears in the scene (i.e. they are class-agnostic). It also means the methods can be used in other contexts, such as vessel traffic monitoring or search and rescue, without significant modifications. This was demonstrated through evaluation on publicly available maritime surveillance datasets with a range of viewpoints, from low in the water (MODD) to high above the sea (SEAGULL).

The image sequences used for evaluation also contained a range of visual challenges related to the maritime domain. The results showed that the proposed methods are robust to camera motion and the dynamic background of the sea. Whilst techniques were developed to mitigate the effects of wake and bright reflections, there is still room for improvement which should be addressed in future work.

The proposed methods were implemented on moderate computing platforms and without extensive optimisation. Full real-time performance (>25Hz) was not reached, but the results obtained indicate that real-time performance is achievable with further effort to optimise the implementations.

A further objective of this work was to evaluate and compare the performance of the proposed methods and others from the literature in the context of a real-world maritime surveillance system using realistic data. Realistic data was obtained through the IPATCH project, which organised the collection of a new maritime sensor dataset based on scenarios developed by experts on pirate behaviour. Evaluation in the context of a real-world system was the focus of Chapter 6. The data collected in the IPATCH trials was used to simulate the real system, which provided insight into how the visual object detection methods supported the higher-level tasks in surveillance.

7.3 Future work

In addition to further experiments and improvements to the proposed algorithms, the investigations in this study have raised many new questions and ideas which would be interesting to explore further:

- Scene context was shown to be a useful enabler for object detection. What other contextual cues can be exploited, in addition to horizon detection and simple rules?
- How can the saliency method incorporate top-down processing in a task-specific way to extract targets more accurately and when the scene is more crowded?
- How can the temporal dimension be incorporated in the saliency approach, beyond basic frame-to-frame tracking?
- How can the network architecture be adapted to better support the horizon and boundary prediction tasks? In this study, the network architectures were left unchanged to isolate the impact of learning multiple tasks but creating separate output branches with additional convolutional layers for each task is common [89, 115].
- Can semi-supervised or unsupervised learning be implemented as an alternative way of addressing the lack of maritime training data?
- New approaches to data augmentation were recently proposed: Mixup [239], sample pairs [99] and between-class learning [211]. They train the model on weighted combinations of two images with the idea that the network can learn smoother class boundary transitions, thus being able to generalise better. This idea would be a logical extension to the work in Chapter 5.
- The predicted horizon and boundary maps contain information which is not currently used. Further research should investigate if this can be exploited in the object detection stage.

To support future work, the MarSemSeg dataset used in Chapter 5 will be made available online. Work on the whole system evaluation presented in Chapter 6 is on-going and a journal paper is under preparation. Synchronising and labelling the data is a time consuming process, but it is hoped that the processed data will be published as part of the journal publication for use by the community and to support future work in this area.

It is also the intention to publish the code for replaying the sensor data to allow other researchers to work with the data more easily and simulate the real conditions on-board. Finally, there is still a lot of work to be done to finish annotating and documenting the complete IPATCH dataset across all three campaigns. This task is left for the next set of PhD students researching visual maritime surveillance...

References

[1] Ablavsky, V. (2003). Background models for tracking objects in water. In Image Process. 2003. ICIP 2003. Proceedings. . . . , pages 125–128.

[2] Achanta, R., Hemamiz, S., Estraday, F., and Süsstrunky, S. (2009). Frequency-tuned salient region detection. 2009 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work. CVPR Work. 2009, (Ic):1597–1604.

[3] Achanta, R. and Sabine, S. (2010). Saliency detection using maximum symmetric surround. In Proc. 2010 IEEE 17th Int. Conf. Image Process., pages 2653–2656.

[4] Ahlberg, J., Berg, A., and Olsson, P.-m. (2019). An Asynchronous Multi-Sensor Tracker for Multiple Maritime Targets. Prepr. https//arxiv.org/submit/2748123.

[5] Ahmad, T., Bebis, G., Nicolescu, M., Nefian, A., and Fong, T. (2016). An edge-less approach to horizon line detection. In Proc. - 2015 IEEE 14th Int. Conf. Mach. Learn. Appl. ICMLA 2015, pages 1095–1102.

[6] Albrecht, T., Tan, T., West, G. A. W., Ly, T., and Moncrieff, S. (2011a). Vision-based attention in maritime environments. In 2011 8th Int. Conf. Information, Commun. Signal Process., pages 1–5.

[7] Albrecht, T., West, G. A. W., and Tan, T. (2011b). Visual Maritime Attention Using Multiple Low-Level Features and Naïve Bayes Classification. In 2011 Int. Conf. Digit. Image Comput. Tech. Appl., pages 243–249.

[8] Alpatov, B. A., Babayan, P. V., and Shubin, N. Y. (2015). Weighted Radon transform for line detection in noisy images. J. Electron. Imaging, 24(2):023023.

[9] Alvarez, J. and Petersson, L. (2016). DecomposeMe: Simplifying ConvNets for End-to-End Learning. arXiv:1606.05426.

[10] Alves, J., Herman, J., and Rowe, N. C. (2004). Robust recognition of ship types from an infrared silhouette. PhD thesis, Thesis Monterey Naval Postgraduate School California USA.

[11] B, E. G., Solmaz, B., and Veysel, Y. (2016). MARVEL: A Large-Scale Image Dataset for Maritime Vessels. In ACCV, pages 165–180.

[12] Badrinarayanan, V., Handa, A., and Cipolla, R. (2015). SegNet: A Deep Convolutional Encoder-Decoder Architecture for Robust Semantic Pixel-Wise Labelling. arXiv:1505.07293.

[13] Badrinarayanan, V., Kendall, A., and Cipolla, R. (2017). Segnet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE Trans. Pattern Anal. Mach. Intell., 39(12):2481–2495.

[14] Bansal, A., Nanduri, A., Castillo, C., Ranjan, R., and Chellappa, R. (2017). UMDFaces: An Annotated Face Dataset for Training Deep Networks. In IEEE Int. Jt. Conf. Biometrics, pages 464–473.

[15] Bao, X., Zinger, S., de With, P. H., and Vinotion, B. V. (2012a). Water Region and Multiple Ship Detection for Port Surveillance.

[16] Bao, X., Zinger, S., Wijnhoven, R. G., and With, P. H. N. D. (2012b). Identification in Port Surveillance. In ACIVS 2012, pages 444–454.

[17] Bar-Shalom, Y., Willett, P. K., and Tian, X. (2011). Tracking and Data Fusion: A Handbook of Algorithms. YBS Publishing.

[18] Berg, A., Ahlberg, J., and Felsberg, M. (2016). Channel Coded Distribution Field Tracking for Thermal Infrared Imagery. In IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work., pages 1248–1256.

[19] Bernardin, K. and Stiefelhagen, R. (2008). Evaluating Multiple Object Tracking Performance: The CLEAR MOT Metrics. EURASIP J. Image Video Process., 2008.

[20] Blackman, S. and Popoli, R. (1999). Design and Analysis of Modern Tracking Systems. Artech House.

[21] Blasch, E. and Valin, P. (2011). Track purity and current assignment ratio for target tracking and identification evaluation. Inf. Fusion (FUSION), 2011 Proc. 14th Int. Conf., (Imm):1–8.

[22] Blasch, E. P., Rice, A., and Yang, C. (2006). Nonlinear tracking evaluation using absolute and relative metrics. Signal Data Process. Small Targets 2006, 6236:62360L.

[23] Bloisi, D. and Iocchi, L. (2009). Argos—A video surveillance system for boat traffic monitoring in Venice. Int. J. Pattern Recognit. ....

[24] Bloisi, D. and Iocchi, L. (2012). Independent Multimodal Background Subtraction. In Proc. Int. Conf. Comput. Model. Objects Present. Images, Fundam., Methods Appl., pages 39–44.

[25] Bloisi, D., Iocchi, L., Fiorini, M., and Graziano, G. (2011). Automatic maritime surveillance with visual target detection. Proc. Int. ..., (c).

[26] Bloisi, D., Iocchi, L., Fiorini, M., and Graziano, G. (2012). Camera based target recognition for maritime awareness. Fusion (FUSION), 2012, pages 1982–1987.

[27] Bloisi, D., Iocchi, L., Pennisi, A., and Tombolini, L. (2015). ARGOS-Venice Boat Classification. In 12th Int. Conf. Adv. Video Signal Based Surveill., pages 1–6.

[28] Bloisi, D., Pennisi, A., and Iocchi, L. (2014). Background modeling in the maritime domain. Mach. Vis. Appl., 25(5):1257–1269.

[29] Bloisi, D., Previtali, F., Pennisi, A., Nardi, D., and Fiorini, M. (2017a). Enhancing Automatic Maritime Surveillance Systems With Visual Information. IEEE Trans. Intell. Transp. Syst., 18(4):824 –833.

[30] Bloisi, D. D., Pennisi, A., and Iocchi, L. (2017b). Parallel multi-modal background modeling. Pattern Recognit. Lett., 96:45–54.

[31] Borghgraef, A., Barnich, O., Lapierre, F., Van Droogenbroeck, M., Philips, W., and Acheroy, M. (2010). An evaluation of pixel-based methods for the detection of floating objects on the sea surface. ... on Adv. ..., 2010.

[32] Borji, A., Cheng, M.-M., Jiang, H., and Li, J. (2014). Salient Object Detection: A Survey. IEEE Trans. image Process., 24(13):5706–5722.

[33] Borji, A., Cheng, M.-M., Jiang, H., and Li, J. (2015). Salient Object Detection: A Benchmark. IEEE Trans. Image Process., 24(12):5706–5722.

[34] Borji, A. and Itti, L. (2015). CAT2000: A Large Scale Fixation Dataset for Boosting Saliency Research. In CVPR 2015 Work. Futur. Datasets.

[35] Bouma, H., de Lange, D.-J. J., Broek, S. P. V. D., Kemp, R. A. W., and Schwering, P. B. W. (2008). Automatic detection of small surface targets with electro-optical sensors in a harbor environment. SPIE Eur. ..., 7114:711402–711402–8.

[36] Bousetouane, F. and Morris, B. (2016). Fast CNN surveillance pipeline for fine-grained vessel classification and detection in maritime scenarios. In 2016 13th IEEE Int. Conf. Adv. Video Signal Based Surveillance, AVSS 2016, pages 242–248.

[37] Bovcon, B. and Kristan, M. (2019). Benchmarking Semantic Segmentation Methods for Obstacle Detection on a Marine Environment. In Comput. Vis. Winter Work., pages 61–70.

[38] Bovcon, B., Mandeljc, R., Perš, J., and Kristan, M. (2018). Stereo obstacle detection for unmanned surface vehicles by IMU-assisted semantic segmentation. Rob. Auton. Syst., 104:1–13.

[39] Broek, S. P. V. D., Bouma, H., den Hollander, R., Veerman, H., Benoist, K., and Schwering, P. B. W. (2014a). Ship recognition for improved persistent tracking with descriptor localization and compact representations. In Huckridge, D. A. and Ebert, R., editors, SPIE ..., volume 9249, page 92490N.

[40] Broek, S. P. V. D., Bouma, H., Veerman, H., Benoist, K., den Hollander, R., and Schwering, P. B. W. (2014b). Recognition of ships for long-term tracking. In Kadar, I., editor, SPIE ..., page 909107.

[41] Brostow, G. J., Fauqueur, J., and Cipolla, R. (2009). Semantic object classes in video: A high-definition ground truth database. Pattern Recognit. Lett., 30(2):88–97.

[42] Bylinskii, Z., Judd, T., Borji, A., Itti, L., Durand, F., Oliva, A., and Torralba, A. (2012). MIT Saliency Benchmark, http://saliency.mit.edu.

[43] Can, T., Karali, A., Aytaç, T., and Karalı, A. O. (2011). Detection and tracking of sea-surface targets in infrared and visual band videos using the bag-of-features technique with scale-invariant feature transform. Appl. Opt.

[44] Cane, T. and Ferryman, J. (2016). Saliency-Based Detection for Maritime Object Tracking. In IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work., pages 1257–1264.

[45] Cane, T. and Ferryman, J. (2018). Evaluating deep semantic segmentation networks for object detection in maritime surveillance. Proc. AVSS 2018 - 2018 15th IEEE Int. Conf. Adv. Video Signal-Based Surveill., pages 1–6.

[46] Caruana, R. (1997). Multitask learning. Mach. Learn., 28(1):41–75.

[47] Čehovin, L., Leonardis, A., and Kristan, M. (2016). Visual Object Tracking Performance Measures Revisited. IEEE Trans. Image Process., 25(3):1261–1274.

[48] Chan, M. T. and Weed, C. (2012). Vessel detection in video with dynamic maritime background. Appl. Imag. Pattern Recognit. ..., (1).

[49] Chen, L. C., Hermans, A., Papandreou, G., Schroff, F., Wang, P., and Adam, H. (2018a). MaskLab: Instance Segmentation by Refining Object Detection with Semantic and Direction Features. In Proc. IEEE Conf. Comput. Vis. Pattern Recognit., pages 4013–4022.

[50] Chen, Y., Chen, X., Zhu, J., Lin, F., and Chen, B. M. (2018b). Development of an autonomous unmanned surface vehicle with object detection using deep learning. Proc. IECON 2018 - 44th Annu. Conf. IEEE Ind. Electron. Soc., 1:5636–5641.

[51] Cheng, M.-M., Mitra, N. J., Huang, X., Torr, P. H. S., and Hu, S.-M. (2015). Global Contrast Based Salient Region Detection. Trans. Pattern Anal. Mach. Intell., 37(3):569–582.

[52] Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., and Schiele, B. (2016). The Cityscapes Dataset for Semantic Urban Scene Understanding. In Proc. IEEE Conf. Comput. Vis. pattern Recognit.

[53] Cruz, G. and Bernardino, A. (2016). Aerial Detection in Maritime Scenarios Using Convolutional Neural Networks. In Int. Conf. Adv. Concepts Intell. Vis. Syst., pages 373–384.

[54] Cruz, G. and Bernardino, A. (2017). Evaluating aerial vessel detector in multiple maritime surveillance scenarios. Ocean. 2017 - Anchorage.

[55] Cruz, G. and Bernardino, A. (2019). Learning Temporal Features for Detection on Maritime Airborne Video Sequences Using Convolutional LSTM. IEEE Trans. Geosci. Remote Sens., pages 1–12.

[56] Dahlkamp, H., Kaehler, A., Stavens, D., Thrun, S., and Bradski, G. (2006). Self-supervised Monocular Road Detection in Desert Terrain. Robot. Sci. Syst., 38.

[57] Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In Int. Conf. Comput. Vis. Pattern Recognit., pages 886–893.

[58] Davis, J. and Goadrich, M. (2006). The relationship between Precision-Recall and ROC curves. In Proc. 23rd Int. Conf. Mach. Learn. - ICML '06, pages 233–240.

[59] Dawkins, M. and Sun, Z. (2014). Tracking nautical objects in real-time via layered saliency detection. In Proc. SPIE 9089, Geospatial InfoFusion Video Anal. IV; Motion Imag. ISR Situational Aware. II, page 908903.

[60] de Villiers, J. and le Roux, F. (2010). Omnidirectional maritime surveillance. Proc. CSIR 3rd Bienn. Conf., pages 1–9.

[61] Dijk, J., Bijl, P., Broek, S. P. V. D., and van Eijk, A. M. (2014). Research topics on EO systems for maritime platforms. SPIE Secur. . . . , 9249:92490M.

[62] Doermann, D. and Mihalcik, D. (2000). Tools and techniques for video performance evaluation. Proc. 15th Int. Conf. Pattern Recognition. ICPR-2000, 4:0–3.

[63] Dollar, P., Wojek, C., Schiele, B., and Perona, P. (2011). Pedestrian Detection:An Evaluation of the State of the Art. IEEE Trans. Pattern Anal. Mach. Intell., 34(4):743–761.

[64] Dusha, D. (2007). Fixed-Wing Attitude Estimation Using Computer Vision Based Horizon Detection. In 12th Aust. Int. Aerosp. Congr., pages 1–19.

[65] Ettinger, S. M., Nechyba, M. C., and Ifju, P. G. (2002). Towards flight autonomy: Vision-based horizon detection for micro air vehicles. Florida Conf., 7(17):617–640.

[66] Ettinger, S. M., Nechyba, M. C., Ifju, P. G., and Waszak, M. (2003). Vision-guided flight stability and control for micro air vehicles. Adv. Robot., 3:2134–2140.

[67] Everingham, M., Van Gool, L., Williams, C. K., Winn, J., and Zisserman, A. (2010). The pascal visual object classes (VOC) challenge. Int. J. Comput. Vis., 88(2):303–338.

[68] Facebook (2017). Facebook Detectron Model Zoo, https://github.com/facebookresearch/Detectron/blob/master/MODEL_ZOO.md.

[69] Fawcett, T. (2006). An introduction to ROC analysis. Pattern Recognit. Lett., 27(8):861–874.

[70] Fefilatyev, S. and Goldgof, D. (2008). Detection and tracking of marine vehicles in video. Pattern Recognition, 2008. ICPR 2008 . . . , pages 0–3.

[71] Fefilatyev, S., Goldgof, D., and Lembke, C. (2010). Tracking Ships from Fast Moving Camera through Image Registration. In Int. Conf. Pattern Recognit., pages 3500–3503.

[72] Fefilatyev, S., Goldgof, D., Shreve, M., and Lembke, C. (2012). Detection and tracking of ships in open sea with rapidly moving buoy-mounted camera system. Ocean Eng., 54:1–12.

[73] Felzenszwalb, P., Girshick, R., McAllester, D., and Ramanan, D. (2010). Object detection with discriminatively trained part-based models. Trans. Pattern Anal. Mach. Intell., 32(9):1627–1645.

[74] Felzenszwalb, P. F. and Huttenlocher, D. P. (2004). Efficient Graph-Based Image Segmentation. Int. J. Comput. Vis., 59(2):167–181.

[75] Ferryman, J. and Shahrokni, A. (2009). PETS2009: Dataset and challenge. In Proc. 12th IEEE Int. Work. Perform. Eval. Track. Surveillance, PETS-Winter 2009.

[76] Frost, D. and Tapamo, J. R. (2013). Detection and tracking of moving objects in a maritime environment using level set with shape priors. EURASIP J. Image Video Process., pages 1–16.

[77] Gallego, A.-J., Pertusa, A., and Gil, P. (2018). Automatic Ship Classification from Optical Aerial Images with Convolutional Neural Networks. Remote Sens., 10(4):511.

[78] Gao, C., Meng, D., Yang, Y., Wang, Y., Zhou, X., and Hauptmann, A. G. (2013). Infrared patch-image model for small target detection in a single image. IEEE Trans. Image Process., 22(12):4996–5009.

[79] Garcia-Garcia, A., Orts-Escolano, S., Oprea, S., Villena-Martinez, V., and Garcia-Rodriguez, J. (2017). A Review on Deep Learning Techniques Applied to Semantic Segmentation. arXiv Prepr. arXiv1704.06857, pages 1–23.

[80] Geiger, A., Lenz, P., and Urtasun, R. (2012). Are we ready for autonomous driving? the KITTI vision benchmark suite. In Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., pages 3354–3361. IEEE.

[81] Gershikov, E., Libe, T., and Kosolapov, S. (2013). Horizon Line Detection in Marine Images: Which Method to Choose? Int. J. Adv. Intell. Syst., 6(2):79–88.

[82] Girshick, R. (2015). Fast R-CNN. In Proc. IEEE Int. Conf. Comput. Vis., pages 1440–1448.

[83] Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., pages 580–587.

[84] Gonzalez, R. C. and Woods, R. E. (2009). Digital image processing. Pearson Education.

[85] Gorji, A. A., Tharmarasa, R., and Kirubarajan, T. (2011). Performance measures for multiple target tracking problems. In 14th Int. Conf. Inf. Fusion, pages 1–8.

[86] Grelsson, B., Felsberg, M., and Isaksson, F. (2015). Highly Accurate Attitude Estimation via Horizon Detection. J. F. Robot., 27.

[87] Hariharan, B., Arbelaez, P., Bourdev, L., Maji, S., and Malik, J. (2011). Semantic Contours from Inverse Detectors. In Int. Conf. Comput. Vis.

[88] Hayder, Z., He, X., and Salzmann, M. (2017). Boundary-aware instance segmentation. In Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017, volume 2017-Janua, pages 587–595.

[89] He, K., Gkioxari, G., Dollar, P., and Girshick, R. (2017). Mask R-CNN. Proc. IEEE Int. Conf. Comput. Vis., pages 2980–2988.

[90] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep Residual Learning for Image Recognition. In Conf. Comput. Vis. Pattern Recognit., pages 770–778.

[91] Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. arXiv:1207.0580, pages 1–18.

[92] Hou, Q., Cheng, M. M., Hu, X., Borji, A., Tu, Z., and Torr, P. (2017). Deeply supervised salient object detection with short connections. In 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017, pages 3202–3212.

[93] Hou, Q., Cheng, M. M., Hu, X., Borji, A., Tu, Z., and Torr, P. (2018). Deeply supervised salient object detection with short connections. Trans. Pattern Anal. Mach. Intell.

[94] Hou, X. and Zhang, L. (2007). Saliency Detection: A Spectral Residual Approach. In Comput. Vis. Pattern Recognition, 2007. CVPR'07. IEEE Conf.

[95] Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. (2017a). Densely Connected Convolutional Networks Gao. In Proc. IEEE Conf. Comput. Vis. pattern Recognit., pages 4700–4708.

[96] Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., Fischer, I., Wojna, Z., Song, Y., Guadarrama, S., and Murphy, K. (2017b). Speed/accuracy trade-offs for modern convolutional object detectors. In Proc. IEEE Conf. Comput. Vis. pattern Recognit., pages 7310–7311.

[97] Huang, L. and Pashler, H. (2007). A Boolean Map Theory of Visual Attention. Psychol. Rev., 114(3):599–631.

[98] (IMO), I. M. O. (1974). International Convention for the Safety of Life at Sea (SOLAS).

[99] Inoue, H. (2018). Data Augmentation by Pairing Samples for Images Classification. arXiv:1801.02929.

[100] International Maritime Organisation (IMO), . (2011). Armed Robbery Against Ships in Waters off the Coast of Somlia: Best Management Practices to Deter Piracy in the Gulf of Aden and Off the Coast of Somalia Developed by the Industry, IMO MSC.1/Circ.1339. Technical report.

[101] Ioffe, S. and Szegedy, C. (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. arXiv:1502.03167.

[102] IPATCH (2017). The IPATCH Project, http://www.ipatchproject.eu.

[103] Itti, L. (2001). Feature combination strategies for saliency-based visual attention systems. J. Electron. Imaging, 10(1):161.

[104] Itti, L. and Koch, C. (2001). Computational modelling of visual attention. Nat. Rev. Neurosci., 2(March):194–203.

[105] Itti, L., Koch, C., and Niebur, E. (1998). A Model of Saliency-Based Visual Attention for Rapid Scene Analysis. IEEE Trans. Pattern Anal. Mach. Intell., 20(11):1254–1259.

[106] Jeong, C. Y., Yang, H., and Moon, K.-D. (2018). Horizon detection in maritime images using scene parsing network. Electron. Lett.

[107] Jiang, H., Wang, J., Yuan, Z., Cheng, M. M., Hu, X., and Zheng, N. (2013). Salient Object Detection: A Discriminative Regional Feature Integration Approach. In Int. Conf. Comput. Vis. Pattern Recognit., pages 2083–2090.

[108] Jiang, Z. and Davis, L. S. (2013). Submodular salient region detection. In Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., pages 2043–2050.

[109] Kadyrov, A., Yu, H., and Liu, H. (2013). Ship Detection and Segmentation Using Image Correlation. 2013 IEEE Int. Conf. Syst. Man, Cybern., pages 3119–3126.

[110] Kaimakis, P. and Tsapatsoulis, N. (2013). Background Modeling Methods for Visual Detection of Maritime Targets. In ARTEMIS '13 Proc. 4th ACM/IEEE Int. Work. Anal. Retr. tracked events motion Imag. stream, pages 67–76.

[111] Kalman, R. E. (1960). A New Approach to Linear Filtering and Prediction Problems. J. Basic Eng., 82(1):35–45.

[112] Kang, W. J., Ding, X. M., Cui, J. W., and Ao, L. (2006). Research on Extraction of Ship Target in Complex Sea-sky Background. J. Phys. Conf. Ser., 48:354–358.

[113] Karchevskiy, M., Ashrapov, I., and Kozinkin, L. (2018). Automatic salt deposits segmentation: A deep learning approach. arXiv:1812.01429.

[114] Kasturi, R., Goldgof, D., Soundararajan, P., Manohar, V., Garofolo, J., Bowers, R., Boonstra, M., Korzhova, V., and Zhang, J. (2009). Framework for performance evaluation of face, text, and vehicle detection and tracking in video: Data, metrics, and protocol. IEEE Trans. Pattern Anal. Mach. Intell., 31(2):319–336.

[115] Kendall, A., Gal, Y., and Cipolla, R. (2018). Multi-Task Learning Using Uncertainty to Weigh Losses for Scene Geometry and Semantics. In CVPR, pages 7482–7491.

[116] Kim, K., Hong, S., Choi, B., and Kim, E. (2018). Probabilistic Ship Detection and Classification Using Deep Learning. Appl. Sci., 8(6):936.

[117] Kim, S. (2013). Analysis of small infrared target features and learning-based false detection removal for infrared search and track. Pattern Anal. Appl., 17:883–900.

[118] Kong, X., Liu, L., Qian, Y., and Cui, M. (2016). Automatic detection of sea-sky horizon line and small targets in maritime infrared imagery. Infrared Phys. Technol., 76:185–199.

[119] Kristan, M., Kenk, V. S., Kovačič, S., and Perš, J. (2016a). Fast Image-Based Obstacle Detection from Unmanned Surface Vehicles. IEEE Trans. Cybern., 46(3):641–654.

[120] Kristan, M., Leonardis, A., Matas, J., Felsberg, M., Pflugfelder, R., Zajc, L., Vojí, T., Bhat, G., Lukežič, A., Eldesokey, A., Fernández, G., García-Martín, A., Iglesias-Arias, A., Aydin Alatan, A., González-García, A., Petrosino, A., Memarmoghadam, A., Vedaldi, A., Muhič, A., He, A., Smeulders, A., Perera, A. G., Li, B., Chen, B., Kim, C., Xu, C., Xiong, C., Tian, C., Luo, C., Sun, C., Hao, C., Kim, D., Mishra, D., Chen, D., Wang, D., Wee, D., Gavves, E., Gundogdu, E., Velasco-Salido, E., Shahbaz Khan, F., Yang, F., Zhao, F., Li, F., Battistone, F., De Ath, G., K S Subrahmanyam, G. R., Bastos, G., Ling, H., Kiani Galoogahi, H., Lee, H., Li, H., Zhao, H., Fan, H., Zhang, H., Possegger, H., Li, H., Lu, H., Zhi, H., Li, H., Lee, H., Jin Chang, H., Drummond, I., Valmadre, J., Spencer Martin, J., Chahl, J., Young Choi, J., Li, J., Wang, J., Qi, J., Sung, J., Johnander, J., Henriques, J., Choi, J., van de Weijer, J., Rodríguez Herranz, J., Martínez, J. M., Kittler, J., Zhuang, J., Gao, J., Grm, K., Zhang, L., Wang, L., Yang, L., Rout, L., Si, L., Bertinetto, L., Chu, L., Che, M., Edoardo Maresca, M., Danelljan, M., Yang, M.-H., Abdelpakey, M., Shehata, M., Kang, M., Lee, N., Wang, N., Miksik, O., Moallem, P., Vicente-Moñivar, P., Senna, P., Li, P., Torr, P., Mariam Raju, P., Ruihe, Q., Wang, Q., Zhou, Q., Guo, Q., Martín-Nieto, R., Krishna Gorthi, R., Tao, R., Bowden, R., Everson, R., Wang, R., Yun, S., Choi, S., Vivas, S., Bai, S., Huang, S., Wu, S., Hadfield, S., Wang, S., Golodetz, S., Ming, T., Xu, T., Zhang, T., Fischer, T., Santopietro, V., Struc, V., Wei, W., Zuo, W., Feng, W., Wu, W., Zou, W., Hu, W., Zhou, W., Zeng, W., Zhang, X., Wu, X., Wu, X.-J., Tian, X., Li, Y., Lu, Y., Wei Law, Y., Wu, Y., Demiris, Y., Yang, Y., Jiao, Y., Li, Y., Zhang, Y., Sun, Y., Zhang, Z., Zhu, Z., Feng, Z.-H., Wang, Z., and He, Z. (2018). The sixth Visual Object Tracking VOT2018 challenge results. In Proc. Eur. Conf. Comput. Vis.

[121] Kristan, M., Matas, J., Leonardis, A., Vojir, T., Pflugfelder, R., Fernandez, G., Nebehay, G., Porikli, F., and Cehovin, L. (2016b). A Novel Performance Evaluation Methodology for Single-Target Trackers. IEEE Trans. Pattern Anal. Mach. Intell., 38(11):2137–2155.

[122] Kristan, M., Per, J., Suli, V., and Kova, S. (2015). A Graphical Model for Rapid Obstacle Image-Map Estimation from Unmanned Surface Vehicles. In ACCV.

[123] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Approximation algorithms for multilevel graph partitioning. In Adv. Neural Inf. Process. Syst., pages 1097—-1105.

[124] Kuhn, H. W. (1955). The Hungarian method for the assignment problem. Nav. Res. Logist. Q., 2(1-2):83–97.

[125] Kumar, N. (2018). Thresholding in salient object detection: a survey. Multimed. Tools Appl., 77(15):19139–19170.

[126] Kuznetsova, A., Rom, H., Alldrin, N., Uijlings, J., Krasin, I., Pont-Tuset, J., Kamali, S., Popov, S., Malloci, M., Duerig, T., and Ferrari, V. (2018). The Open Images Dataset V4: Unified image classification, object detection, and visual relationship detection at scale. arXiv:1811.00982, pages 1–20.

[127] Larsson, F. and Felsberg, M. (2011). Using Fourier descriptors and spatial models for traffic sign recognition. In Scand. Conf. image Anal., pages 238–249.

[128] Laurinen, M. (2016). Remote and Autonomous Ships: The next steps (http://www.rolls-royce.com/~/media/Files/R/Rolls-Royce/documents/customers/marine/ship-intel/aawa-whitepaper-210616.pdf). Technical report.

[129] Leal-Taixé, L., Milan, A., Reid, I., Roth, S., and Schindler, K. (2015). MOTChallenge 2015: Towards a Benchmark for Multi-Target Tracking. arXiv:1504.01942, pages 1–15.

[130] Lerczak, J. A. and Hobbs, R. C. (1998). Calculating Sighting Distances From Angular Readings During Shipboard, Aerial, and Shore-Based Marine Mammal Surveys. Mar. Mammal Sci., 14(3):590–598.

[131] Li, X., Flohr, F., Yang, Y., Xiong, H., Braun, M., Pan, S., Li, K., and Gavrila, D. M. (2016). A new benchmark for vision-based cyclist detection. In IEEE Intell. Veh. Symp. Proc., volume 2016-Augus, pages 1028–1033. IEEE.

[132] Li, Y., Bian, X., Chang, M. C., Wen, L., and Lyu, S. (2018). Pixel Offset Regression (POR) for Single-shot Instance Segmentation. In Proc. 15th IEEE Int. Conf. Adv. Video Signal-Based Surveill.

[133] Li, Y., Hua, W., Guo, C., Gu, H., Kang, J., and Chen, X. (2008). Waterfront surveillance and trackability. Mach. Vis. Appl., 19(5-6):291–300.

[134] Lim, K. H., Seng, K. P., and Siew, W. (2009). Vision-based Lane-Vehicle Detection and Tracking. In AIP Conf. Proc., volume 1174, pages 157—-171.

[135] Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollar, P., and Zitnick, C. L. (2014). Microsoft COCO: Common objects in context. In Eur. Conf. Comput. Vis., pages 740–755.

[136] Liu, H., Javed, O., Taylor, G., Cao, X., and Haering, N. (2008). Omni-directional surveillance for unmanned water vehicles. In Eighth Int. Work. Vis. Surveill. - VS2008.

[137] Liu, R., Lehman, J., Molino, P., Such, F. P., Frank, E., Sergeev, A., and Yosinski, J. (2018a). An Intriguing Failing of Convolutional Neural Networks and the CoordConv Solution. In NIPS, pages 1–12.

[138] Liu, S., Ding, W., Liu, C., Liu, Y., Wang, Y., and Li, H. (2018b). ERN: Edge Loss Reinforced Semantic Segmentation Network for Remote Sensing Images. Remote Sens., 10(9):1339.

[139] Lo, S.-Y., Hang, H.-M., Chan, S.-W., and Lin, J.-J. (2018). Efficient Dense Modules of Asymmetric Convolution for Real-Time Semantic Segmentation. arXiv:1809.06323.

[140] Long, J., Shelhamer, E., and Darrell, T. (2015). Fully Convolutional Networks for Semantic Segmentation. In Proc. IEEE Conf. Comput. Vis. pattern Recognit., pages 3431–3440.

[141] Loomans, M. J., Wijnhoven, R. G., and de With, P. H. (2013). Robust automatic ship tracking in harbours using active cameras. ICIP, pages 4117–4121.

[142] Luo, Q., Khoshgoftaar, T. M., Folleco, A., and Raton, B. (2006). Classification of Ships in Surveillance Video. pages 432–437.

[143] Lynch, D. K. (2008). Visually discerning the curvature of the Earth. Appl. Opt., 47(34):H39.

[144] Makantasis, K., Doulamis, A., and Doulamis, N. (2013). Vision-based Maritime Surveillance System using Fused Visual Attention Maps and Online Adaptable Tracker. In 14th Int. Work. Image Anal. Multimed. Interact. Serv., pages 1–4.

[145] Makantasis, K., Protopapadakis, E., Doulamis, A., and Matsatsinis, N. (2015). Semi-supervised vision-based maritime surveillance system using fused visual attention maps. In Multimed. Tools Appl., volume 75, pages 15051–15078.

[146] Marie, V., Bechar, I., and Bouchara, F. (2018). Real-time maritime situation awareness based on deep learning with dynamic anchors. In Adv. Video Signal Based Surveill. (AVSS), IEEE Int. Conf.

[147] Mehta, S., Rastegari, M., Caspi, A., Shapiro, L., and Hajishirzi, H. (2018). ESPNet: Efficient Spatial Pyramid of Dilated Convolutions for Semantic Segmentation. arXiv:1803.06815.

[148] Milan, A., Leal-Taixe, L., Reid, I., Roth, S., and Schindler, K. (2016). MOT16: A Benchmark for Multi-Object Tracking. arXiv Prepr. arXiv1603.00831, pages 1–12.

[149] Moosbauer, S., Daniel, K., Jens, J., Teutsch, M., and Gmbh, H. O. (2019). A Benchmark for Deep Learning Based Object Detection in Maritime Environments. In IEEE Conf. Comput. Vis. Pattern Recognit. Work.

[150] Munkres, J. (1957). Algorithms for the Assignment and Transportation Problems. J. Soc. Ind. Appl. Math., 5(1):32–38.

[151] Nawaz, T., Poiesi, F., and Cavallaro, A. (2014). Measures of effective video tracking. IEEE Trans. Image Process., 23(1):376–388.

[152] Neto, A. M., Victorino, A. C., Fantoni, I., and Zampieri, D. E. (2011). Robust horizon finding algorithm for real-time autonomous navigation based on monocular vision. In IEEE Conf. Intell. Transp. Syst. Proceedings, ITSC, pages 532–537.

[153] Oceans Beyond Piracy, . (2014). The State of Maritime Piracy 2014. http://dx.doi.org/10.18289/OBP.2015.001. Technical report.

[154] Oksuz, K., Cam, B. C., Akbas, E., and Kalkan, S. (2018). Localization recall precision (LRP): A new performance metric for object detection. Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), 11211 LNCS:521–537.

[155] Oreifej, O., Lobo, N., and Shah, M. (2011). Horizon constraint for unambiguous UAV navigation in planar scenes. Proc. - IEEE Int. Conf. Robot. Autom., pages 1159–1165.

[156] Osborne, C., Cane, T., Nawaz, T., and Ferryman, J. (2015). Temporally stable feature clusters for maritime object tracking in visible and thermal imagery. In AVSS 2015 - 12th IEEE Int. Conf. Adv. Video Signal Based Surveill.

[157] Otsu, N. (1979). A Threshold Selection Method from Gray-Level Histograms. IEEE Trans. Syst. Man Cybern., SMC-9(1):62–66.

[158] Park, J., Nam, K., and Joo, J. (2011). A Partially Occluded Sea-Sky Line Detection Algorithm. Image Process. Comput. Vision, Pattern ..., 2.

[159] Paszke, A., Chaurasia, A., Kim, S., and Culurciello, E. (2016). ENet: A Deep Neural Network Architecture for Real-Time Semantic Segmentation. arXiv:1606.02147.

[160] Patino, L., Cane, T., Vallee, A., and Ferryman, J. (2016). PETS 2016: Dataset and Challenge. In IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work.

[161] Patino, L. and Ferryman, J. (2014). PETS 2014: Dataset and challenge. 11th IEEE Int. Conf. Adv. Video Signal-Based Surveillance, AVSS 2014, pages 355–360.

[162] Patino, L., Nawaz, T., Cane, T., and Ferryman, J. (2017). PETS 2017: Dataset and Challenge. In IEEE Conf. Comput. Vis. Pattern Recognit. Work., volume 2017-July, pages 2126–2132.

[163] Pires, N., Guinet, J., and Dusch, E. (2010). ASV: An innovative automatic system for maritime surveillance (Un systeme automatique innovant de surveillance maritime). Navigation, 58(232).

[164] Prasad, D. K., Prasath, C. K., Rajan, D., Rachmawati, L., Rajabally, E., and Quek, C. (2018a). Object Detection in a Maritime Environment: Performance Evaluation of Background Subtraction Methods. IEEE Trans. Intell. Transp. Syst., PP:1–16.

[165] Prasad, D. K., Prasath, C. K., Rajan, D., Rachmawati, L., Rajabaly, E., and Quek, C. (2016). Challenges in video based object detection in maritime scenario using computer vision.

[166] Prasad, D. K., Rajan, D., Prasath, C. K., Rachmawati, L., Rajabally, E., and Quek, C. (2017a). MSCM-LiFe: Multi-scale cross modal linear feature for horizon detection in maritime images. IEEE Reg. 10 Annu. Int. Conf. Proceedings/TENCON, pages 1366–1370.

[167] Prasad, D. K., Rajan, D., and Quek, C. (2018b). Are object detection assessment criteria ready for maritime computer vision? arXiv:1809.04659v1, pages 1–9.

[168] Prasad, D. K., Rajan, D., Rachmawati, L., Rajabally, E., and Quek, C. (2017b). Video Processing From Electro-Optical Sensors for Object Detection and Tracking in a Maritime Environment : A Survey. IEEE Trans. Intell. Transp. Syst., pages 1–24.

[169] Purohit, I. K., Islam, M. N., Asari, V., and Karim, M. A. (2008). Target detection using adaptive progressive thresholding based shifted phase-encoded fringe-adjusted joint transform correlator. Int. J. ..., 23529:277–282.

[170] Qi, B., Wu, T., He, H., and Hu, T. (2011). Real-time detection of small surface objects using weather effects. Comput. Vision–ACCV 2010.

[171] Redmon, J. (2019). Darknet Neural Network Library, https://pjreddie.com/darknet/yolo.

[172] Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. In IEEE Conf. Comput. Vis. pattern Recognit., pages 779–788.

[173] Redmon, J. and Farhadi, A. (2017). YOLO9000: Better, faster, stronger. IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017, pages 6517–6525.

[174] Redmon, J. and Farhadi, A. (2018). YOLOv3: An Incremental Improvement. arXiv:1804.02767.

[175] Ren, L., Ran, X., Peng, J., and Shi, C. (2016). Saliency Detection for Small Maritime Target Using Singular Value Decomposition of Amplitude Spectrum. IETE Tech. Rev., 34(6):631–641.

[176] Ren, L., Shi, C., and Ran, X. (2012). Target detection of maritime search and rescue: Saliency accumulation method. In Proc. - 2012 9th Int. Conf. Fuzzy Syst. Knowl. Discov. FSKD 2012, pages 1972–1976.

[177] Ren, L., Shi, C. J., and Ran, X. (2011). Target detection in maritime search and rescue using SVD and frequency domain characteristics. Proc. - Int. Conf. Mach. Learn. Cybern., 2:556–560.

[178] Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In Adv. Neural Inf. Process. Syst., pages 91—-99.

[179] Rhodes, B. J., Bomberger, N. A., Freyman, T. M., Kreamer, W., Kirschner, L., L'Italien, A. C., Mungovan, W., Stauffer, C., Stolzar, L., Waxman, A. M., and Seibert, M. (2007). SeeCoast: Persistent Surveillance and Automated Scene Understanding for Ports and Coastal Areas. In Def. Transform. Net-Centric Syst., volume 6578, page 65781M. SPIE.

[180] Ribeiro, R. (2017). The SEAGULL dataset, http://vislab.isr.ist.utl.pt/seagull-dataset.

[181] Robert-Inacio, F., Raybaud, A., and Clement, E. (2007). Multispectral target detection and tracking for seaport video surveillance. Proc. IVS . . . , (December):169–174.

[182] Romera, E., Alvarez, J. M., Bergasa, L. M., and Arroyo, R. (2017). Efficient ConvNet for real-time semantic segmentation. In IEEE Intell. Veh. Symp., pages 1789–1794.

[183] Romera, E., Alvarez, J. M., Bergasa, L. M., and Arroyo, R. (2018). ERFNet: Efficient Residual Factorized ConvNet for Real-time Semantic Segmentation. IEEE Trans. Intell. Transp. Syst., 19(1):263–272.

[184] Ronneberger, O., Fischer, P., and Brox, T. (2015). U-Net: Convolutional networks for biomedical image segmentation. In Int. Conf. Med. image Comput. Comput. Interv., pages 234–241.

[185] Rother, C., Kolmogorov, V., and Blake, A. (2004). Grabcut: Interactive foreground extraction using iterated graph cuts. ACM Trans. Graph., 23(3):309–314.

[186] Rothrock, R. L. and Drummond, O. E. (2000). Performance Metrics for Multiple-Sensor, Multiple-Target Tracking. In Signal Data Process. Small Targets 2000, volume 4048, pages 521–531.

[187] Ruder, S. (2017). An Overview of Multi-Task Learning in Deep Neural Networks. arXiv:1706.05098, (May).

[188] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. Int. J. Comput. Vis., 115(3):211–252.

[189] Sadhu, T., Albu, A. B., Hoeberechts, M., Wisernig, E., and Wyvill, B. (2016). Obstacle detection for image-guided surface water navigation. Proc. - 2016 13th Conf. Comput. Robot Vision, CRV 2016, pages 45–52.

[190] Sagonas, C., Tzimiropoulos, G., Zafeiriou, S., and Pantic, M. (2013). 300 faces in-the-wild challenge: The first facial landmark Localization Challenge. In Proc. IEEE Int. Conf. Comput. Vis., pages 397–403.

[191] Salton, G. and McGill, M. J. (1986). Introduction to modern information retrieval. McGraw-Hill.

[192] Samama, A. (2010). Innovative video analytics for maritime surveillance. Waterside Secur. Conf. (WSS), 2010 ..., pages 1–8.

[193] Sato, Y. and Ishii, H. (1998). Study of a collision-avoidance system for ships. Control Eng. Pract., 6(9):1141–1149.

[194] Schwering, P. B. W., Broek, S. P. V. D., and Iersel, M. V. (2007). EO system concepts in the littoral. Def. ..., 6542:1–12.

[195] Seibert, M. (2006). SeeCoast port surveillance. Def. ..., 6204:18–19.

[196] Senior, A., Hampapur, A., Tian, Y.-L., Brown, L., Pankanti, S., and Bolle, R. (2006). Appearance models for occlusion handling. Image Vis. Comput., 24(11):1233–1243.

[197] Shao, Z., Wu, W., Wang, Z., Du, W., and Li, C. (2018). SeaShips: A Large-Scale Precisely Annotated Dataset for Ship Detection. IEEE Trans. Multimed., 20(10):2593–2604.

[198] Shmatko, O., Aleksiyev, V., and Dong, L. (2018). An Algorithm for Sea-Sky Line Detection Under Visible Sea Image. Adv. Inf. Syst., 2(4):128–135.

[199] Simonyan, K. and Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv Prepr. arXiv1409.1556, pages 1–14.

[200] Smith, A. and Teal, M. (1999). Identification and Tracking of Maritime Objects in Near-Infrared Image Sequences. In Image Process. Its ....

[201] Sobral, A., Bouwmans, T., and Zahzah, E. H. (2015). Double-constrained RPCA based on saliency maps for foreground detection in automated maritime surveillance. In AVSS 2015 - 12th IEEE Int. Conf. Adv. Video Signal Based Surveill.

[202] Socek, D., Culibrk, D., and Marques, O. (2005). A hybrid color-based foreground object detection method for automated marine surveillance. Adv. Concepts ..., pages 340–347.

[203] Souvenir, R., Wright, J., and Pless, R. (2005). Spatio-temporal detection and isolation: results on PETS 2005 DataSets. Urbana.

[204] Stiefelhagen, R., Bernardin, K., Bowers, R., Garofolo, J., Mostefa, D., and Soundararajan, P. (2006). The CLEAR 2006 Evaluation. In Int. Eval. Work. Classif. Events, Act. Relationships, pages 1–44.

[205] Sumimoto, T., Kuramoto, K., Okada, S., Miyauchi, H., Imade, M., Yamamoto, H., and Arvelyna, Y. (2000). Detection of a particular object from environmental images under various conditions. ... , 2000. ISIE 2000. ....

[206] Sumimoto, T., Kuramoto, K., Okada, S., Miyauchi, H., Imade, M., Yamamoto, H., and Kunishi, T. (1994). Machine vision for detection of the rescue target in the marine casualty. In 20th Int. Conf. Ind. Electron. Control Instrumentation, 1994. IECON '94., pages 723–726.

[207] Sun, Y. and Fu, L. (2018). Coarse-Fine-Stitched: A Robust Maritime Horizon Line Detection Method for Unmanned Surface Vehicle Applications. Sensors, 18(9):2825.

[208] Szpak, Z. L. and Tapamo, J. R. (2011). Maritime surveillance: Tracking ships inside a dynamic background using a fast level-set. Expert Syst. Appl., 38(6):6669–6680.

[209] Tan, S., Piepmeier, J. A., and Kriebel, D. L. (2012). A Computer Vision System for Monitoring Vessel Motion in Conjunction with Vessel Wake Measurements. Signals, Syst. ..., di:1830–1834.

[210] Tangstad, E. (2017). Visual Detection of Maritime Vessels. PhD thesis.

[211] Tokozume, Y., Ushiku, Y., and Harada, T. (2017). Between-class Learning for Image Classification. In CVPR 2017, pages 5486–5494.

[212] Tran, T. H. and Le, T. L. (2016). Vision based boat detection for maritime surveillance. In Int. Conf. Electron. Information, Commun. ICEIC 2016.

[213] Tsai, R. (1987). Versatile Camera Calibration Techniaue for High-Accuracy 3D Machine Vision Metrology. Robot. Autom., (4).

[214] Uhrig, J., Rehder, E., Fröhlich, B., Franke, U., and Brox, T. (2018). Box2Pix: Single-Shot Instance Segmentation by Assigning Pixels to Object Boxes. In IEEE Intell. Veh. Symp. Proc., pages 292–299.

[215] UK Government Home Office, . (2011). i-LIDS Dataset.

[216] Ulman, V., Maška, M., Magnusson, K. E. G., Ronneberger, O., Haubold, C., Harder, N., Matula, P., Matula, P., Svoboda, D., Radojevic, M., Smal, I., Rohr, K., Jaldén, J., Blau, H. M., Dzyubachyk, O., Lelieveldt, B., Xiao, P., Li, Y., Cho, S.-Y., Dufour, A. C., Olivo-Marin, J.-C., Reyes-Aldasoro, C. C., Solis-Lemus, J. A., Bensch, R., Brox, T., Stegmaier, J., Mikut, R., Wolf, S., Hamprecht, F. A., Esteves, T., Quelhas, P., Demirel, Ö., Malmström, L., Jug, F., Tomancak, P., Meijering, E., Muñoz-Barrutia, A., Kozubek, M., and Ortiz-de Solorzano, C. (2017). An objective comparison of cell-tracking algorithms. Nat. Methods, 14(12):1141–1152.

[217] Viola, P. and Jones, M. (2001). Rapid Object Detection using a Boosted Cascade of Simple Features. In Int. Conf. Comput. Vis. Pattern Recognit., pages 511–518.

[218] Voles, P., Smith, A., and Teal, M. (2000). Nautical Scene Segmentation Using Variable Size Image Windows and Feature Space Reclustering. Comput. Vision—ECCV 2000, (i):324–335.

[219] Voles, P., Teal, M., and Sanderson, J. (1999). Target Identification in a Complex Maritime Scene. In IEE Colloq. Motion Anal. Track., pages 15/1–15/4.

[220] Wang, B., Su, Y., and Wan, L. (2016a). A sea-sky line detection method for unmanned surface vehicles based on gradient saliency. Sensors (Switzerland), 16(4).

[221] Wang, D. and Chen, W. (2008). The Study on Ship-Flow Analysis and Counting System in a Specific Sea-Area Based on Video Processing. ... Signal Process. 2008. ..., pages 655–658.

[222] Wang, D., Zhang, C., Cheng, H., Shang, Y., and Mei, L. (2016b). SPID: Surveillance Pedestrian Image Dataset and Performance Evaluation for Pedestrian Detection. In Asian Conf. Comput. Vis., pages 463–477.

[223] Wang, J., Jiang, H., Yuan, Z., Cheng, M. M., Hu, X., and Zheng, N. (2017). Salient Object Detection: A Discriminative Regional Feature Integration Approach. Int. J. Comput. Vis., 123:251–268.

[224] Wang, W., Fu, Y., Dong, F., and Li, F. (2019). Semantic segmentation of remote sensing ship image via a convolutional neural networks model. IET Image Process., 13(6):1016–1022.

[225] Wanli, O., Loy, C., Lin, D., Li, H., and Xiong, Y. (2018). WIDER Face and Person Challenge.

[226] Wei, Z.-Y., Lee, D.-J., Jilk, D., and Schoenberger, R. (2007). Motion projection for floating object detection. Adv. Vis. Comput., pages 152–161.

[227] Westall, P., O'Shea, P., Ford, J. J., and Hrabar, S. (2009). Improved maritime target tracker using colour fusion. ... Comput. Simulation, ..., pages 230–236.

[228] Wijnhoven, R. G., van Rens, K., Jaspers, E. G. T., and de With, P. H. (2010). Online Learning for Ship Detection in Maritime Surveillance. In 31th Symp. Inf. Theory Benelux, pages 73–80.

[229] Wren, C. R. and Porikli, F. (2005). Waviz: Spectral Similarity for Object Detection Introduction Cyclostationarity Using Spectral Similarity. In IEEE Int. Work. Perform. Eval. Track. Surveill.

[230] Xie, S. and Tu, Z. (2015). Holistically-Nested Edge Detection. In IEEE Int. Conf. Comput. Vis. (CVPR)2, pages 1395–1403.

[231] Xu, Z., Yang, W., Meng, A., Lu, N., Huang, H., Ying, C., and Huang, L. (2018). Towards end-to-end license plate detection and recognition: A large dataset and baseline. In ECCV 2018.

[232] Yamamoto, K., Yamada, K., Kiriya, N., and Matsukura, H. (1999). Optical sensing and image processing to detect a life raft. ... Remote Sens. ..., 1:467–469.

[233] Yaman, C. and Asari, V. (2007). Long-range target classification in a cluttered environment using multi-sensor image sequences. ... Sp. Technol. 2007. RAST'07. 3rd ..., pages 304–308.

[234] Yang, J., Price, B., Cohen, S., Lee, H., and Yang, M. H. (2016). Object contour detection with a fully convolutional encoder-decoder network. In Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., volume 2016-Decem, pages 193–202.

[235] Yao, Z. (2013). Small target detection under the sea using multi-scale spectral residual and maximum symmetric surround. Proc. - 2013 10th Int. Conf. Fuzzy Syst. Knowl. Discov. FSKD 2013, pages 241–245.

[236] Yu, C., Wang, J., Peng, C., Gao, C., Yu, G., and Sang, N. (2018). Learning a Discriminative Feature Network for Semantic Segmentation. In Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., volume 1, pages 1857–1866.

[237] Yu, F. and Koltun, V. (2015). Multi-Scale Context Aggregation by Dilated Convolutions. arXiv:1511.07122.

[238] Zack, G. W., Rogers, W. E., and Latt, S. A. (1977). Automatic measurement of sister chromatid exchange frequency. J. Histochem. Cytochem., 25(7):741–753.

[239] Zhang, H., Cisse, M., Dauphin, Y. N., and Lopez-Paz, D. (2017). mixup: Beyond Empirical Risk Minimization. arXiv:1710.09412, pages 1–13.

[240] Zhang, J. and Sclaroff, S. (2013). Saliency Detection: A Boolean Map Approach. 2013 IEEE Int. Conf. Comput. Vis., (Iccv):153–160.

[241] Zhang, J. and Sclaroff, S. (2015). Exploiting Surroundedness for Saliency Detection: A Boolean Map Approach. IEEE Trans. Pattern Anal. Mach. Intell., 38(5):889 – 902.

[242] Zhang, M. M., Choi, J., Daniilidis, K., Wolf, M. T., and Kanan, C. (2015). VAIS: A dataset for recognizing maritime imagery in the visible and infrared spectrums. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work., 2015-Octob:10–16.

[243] Zhang, S., Qi, Z., and Zhang, D. (2009). Ship Tracking Using Background Subtraction and Inter-frame Correlation. pages 1–4.

[244] Zhang, Z. and Wong, K. H. (2013). A computer vision based sea search method using Kalman filter and CAMSHIFT. ... Electr. Electron. Comput. ..., pages 188–193.

[245] Zhao, H., Qi, X., Shen, X., Shi, J., and Jia, J. (2018). ICNet for Real-Time Semantic Segmentation on High-Resolution Images. arXiv:1704.08545v2.

[246] Zhao, X., Ding, W., Liu, C., and Li, H. (2017). Haze removal for unmanned aerial vehicle aerial video based on spatial-temporal coherence optimisation.

[247] Zhengyou Zhang (1999). Flexible camera calibration by viewing a plane from unknown orientations. In Proc. Seventh IEEE Int. Conf. Comput. Vis., volume 1, pages 666–673.

[248] Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., and Torralba, A. (2017). Scene parsing through ADE20K dataset. In Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017, pages 5122–5130.

[249] Zou, Z., Shi, Z., Guo, Y., and Ye, J. (2019). Object Detection in 20 Years: A Survey. arXiv:1905.05055.

Appendix A

Data Augmentations

Below, the implementations of the seven data augmentations used in Section 5.4.2 are described. In the following definitions, $\lfloor x \rceil_a^b$ denotes that the value of x is clamped between a and b, and $\lfloor x \rceil$ denotes integer rounding.

Brightness: Brightness changes are implemented by scaling the lightness channel of the image in HLS colourspace. The image is converted from RGB to HLS and the lightness channel, L, is scaled with a scale factor f^b:

$$L' = \left\lfloor f^b L \right\rceil_0^{255} \tag{A.1}$$

where f^b is drawn from a uniform distribution $f^b \sim \mathscr{U}\left(f^b_{min}, f^b_{max}\right)$. In this work, f^b_{min} and f^b_{max} were set to 0.8 and 1.2, respectively. The H, L′ and S channels are then converted back to RGB colourspace.

Contrast: Contrast changes are implemented by scaling each channel of the image by a scale factor f^c:

$$C' = \left\lfloor f^c C \right\rceil_0^{255} \tag{A.2}$$

where f^c is drawn from a uniform distribution $f^c \sim \mathscr{U}\left(f^c_{min}, f^c_{max}\right)$. The same scale factor is applied to all channels. In this work, f^c_{min} and f^c_{max} were set to 0.8 and 1.2, respectively.

Colour: Colour variations are implemented by applying gamma adjustment to a randomly selected subset, S, of the RGB channels:

$$C' = 255\left(\frac{C}{255}\right)^{\gamma}, \quad \text{for } C \in S, \text{ where } S \subseteq \{R, G, B\} \tag{A.3}$$

where γ is drawn from a uniform distribution $\gamma \sim \mathscr{U}\left(\gamma_{min}, \gamma_{max}\right)$. In this work, γ_{min} and γ_{max} were set to 0.8 and 1.2, respectively.

Rotation: Rotation is implemented by applying an affine transformation, A, to the image which rotates around the image centre (x_c, y_c):

$$A = \begin{bmatrix} \cos\theta & \sin\theta & (1-\cos\theta)x_c - \sin\theta\, y_c \\ -\sin\theta & \cos\theta & \sin\theta\, x_c + (1-\cos\theta)y_c \end{bmatrix} \tag{A.4}$$

where θ is drawn from a discrete uniform distribution $\theta \sim \mathscr{U}\{\theta_{min}, \theta_{max}\}$. In this work, θ_{min} and θ_{max} were set to -5 and 5 degrees, respectively. Undefined pixels in the image are set to black and corresponding pixels in the class label image are filled with the 'Void' class.

Blur: Gaussian blur is applied to the image with standard deviation, σ^b, drawn from a uniform distribution $\sigma^b \sim \mathscr{U}\left(\sigma^b_{min}, \sigma^b_{max}\right)$. In this work, σ^b_{min} and σ^b_{max} were set to 0 and 1.3, respectively. The size of the Gaussian kernel, k, is set according to:

$$k = \max\left(2\left\lfloor\frac{3.3\sigma^b}{2} - 0.5\right\rceil, 3\right) \tag{A.5}$$

Noise: Noise is implemented as additive Gaussian noise by adding random values to each pixel in each image channel:

$$C' = [C + \epsilon]_0^{255} \tag{A.6}$$

where $\epsilon \sim \mathscr{N}\left(0, (\sigma^n)^2\right)$ and σ^n is drawn from a uniform distribution $\sigma^n \sim \mathscr{U}\left(\sigma^n_{min}, \sigma^n_{max}\right)$. In this work, σ^n_{min} and σ^n_{max} were set to 0 and 3, respectively.

Compression artefacts: Compression artefacts are created by applying JPEG compression to the image with randomly selected strengths. Higher strengths remove more high

spatial frequency components leading to lower image quality. In the implementation, the images are saved as JPEG files with different compression strengths, p, and then reloaded 'on the fly'. p is expressed as a percentage, and is drawn from a uniform distribution $p \sim \mathscr{U}\left(p_{min}, p_{max}\right)$. In this work, p_{min} and p_{max} were set to 0 and 40, respectively.