Evaluating deep semantic segmentation networks for object detection in maritime surveillance

Evaluating deep semantic segmentation networks for object detection in maritime surveillance

Tom Cane and James Ferryman
University of Reading, Department of Computer Science, SMPCS
Polly Vacher Building, Whiteknights, Reading RG6 6AY, UK
{t.cane, j.m.ferryman}@reading.ac.uk

Abstract

Maritime surveillance is important for applications in safety and security, but the visual detection of objects in maritime scenes remains challenging due to the diverse and unconstrained nature of such environments, and the need to operate in near real-time. Recent work on deep neural networks for semantic segmentation has achieved good performance in the road/urban scene parsing task. Driven by the potential application in autonomous vehicle navigation, many of the architectures are designed to be fast and lightweight. In this paper, we evaluate semantic segmentation networks in the context of an object detection system for maritime surveillance. Using data from the ADE20k scene parsing dataset, we train a selection of recent semantic segmentation network architectures to compare their performance on a number of publicly available maritime surveillance datasets.

1. Introduction

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. The combination of off-the-shelf video cameras and modern computer vision techniques offers rich visual information at an affordable cost. However, there are a number of visual sensing challenges associated with operating in the maritime domain: dynamic background, reflections, the large variety of objects which may be encountered, and extreme environmental conditions. Due to the unconstrained nature of the domain, maritime surveillance may take place from a number of different viewpoints, ranging from a camera mounted on a small surface vehicle close to the waterline, to static shore-based cameras, to aerial surveillance from planes or drones. It is therefore desirable that any approach is able to generalise to different viewpoints and accommodate substantial differences in object appearance. In addition, methods should be robust to large camera motions and be able to operate in real-time. In this paper, we evaluate the use of deep semantic segmentation networks as part of a maritime visual object detection system concept (Fig. 1) and compare their performance on sequences from publicly available maritime surveillance datasets. Our objective is to see how well this approach can cope with the challenges of maritime environments based on limited training data.

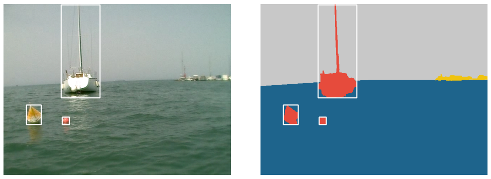


Figure 1: Concept of semantic segmentation for object detection.

2. Related work

Recent approaches for maritime object detection [2, 3, 5, 7, 11, 12, 13, 20, 22, 23] include object classification, background modelling, and saliency based methods. Detection based on classifiers using Haar [2], HOG [12] or CNN [3, 7] features require substantial training data and are prone to overfitting on specific subsets of maritime objects. Modelling maritime backgrounds (i.e. sea and sky) using colour [13], texture [23] or graphical [11] models avoids overfitting for specific object classes, but may fail to generalise well to different environmental or lighting conditions. Saliency-based methods [5, 20, 22] are model-free and robust to these challenges, but are unable to distinguish between maritime objects of interest and any other salient regions of the scene (such as land).

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 [8]. Deep semantic segmentation networks represent the

978-1-5386-9294-3/18/$31.00 ©2018

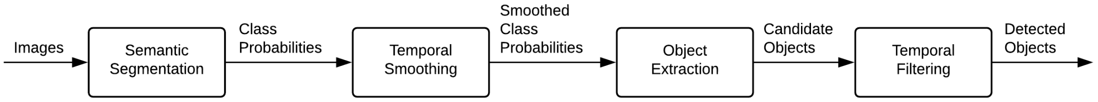


Figure 2: Block diagram presenting the different stages of the proposed approach.

current state-of-the-art in road/urban scene parsing competitions [4, 6], even when trained with relatively small datasets (100s of images) [1]. This makes them an attractive candidate for use in the maritime domain, where there is very little publicly available annotated data. Driven by the potential application in autonomous vehicle navigation, many of the architectures [14, 15] are also designed with speed and memory consumption in mind so that they can run in real time on low-power hardware.

3. System Overview

We propose a simple object detection system (Fig. 2) which exploits the characteristics of deep semantic segmentation networks to address the challenges of maritime environments. 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 class probability distributions are smoothed over time by taking the mean of a sliding window over 3 frames. A binary map is created for each class by selecting pixels where that class is the maximum probability and then morphologically filtering to remove small, disconnected regions and fill small holes. We apply the opening operation with a kernel of size 5, followed by closing with a kernel of size 10. Candidate regions are extracted from the binary maps by labelling connected components and computing bounding boxes. To remove transient detections, candidate regions are matched frame-to-frame based on their overlap ratio, and their positions are filtered with a Kalman filter. Detections which are stable for more than 5 frames are output for each frame as a set of bounding boxes, each with an associated class label.

4. Semantic segmentation networks

4.1. Network architectures

We select three semantic segmentation networks from recent literature which have been designed to be efficient, motivated by use in real-time applications such as autonomous driving. Two networks – ENet [15] and ESPNet [14] – report performance speeds >20 FPS. The third network (SegNet [1]) runs slower, but is included in the comparison to assess the trade-off between accuracy and speed. All three networks obtain similar accuracy performance on benchmark datasets such as CamVid [4] and CityScapes [6] (see Table 1). The networks are fully convolutional and do not rely on post-processing of the network output (e.g. using CRF refinement [8]) 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.

Network	Params	FPS	mIOU	Class Balancing	Optimisation
SegNet [1]	29.5M	1.6	57.0	Median Frequency	SGD; lr: 0.001, momentum: 0.9; weight decay: 0.0005
ENet [15]	0.36M	21.6	58.3	$w_i = \frac{1}{\ln(1.02 + p_i)}$	ADAM; βs: 0.9, 0.999; lr: 0.005, γ: 0.1; weight decay: 0.0002
ESPNet [14]	0.36M	27.4	60.3	$w_i = \frac{1}{\ln(1.02 + p_i)}$	ADAM; βs: 0.9, 0.999; lr: 0.0005, γ: 0.5; weight decay: 0.0005

Table 1: Semantic segmentation networks. Values for no. of parameters, FPS and mIOU are for the CityScapes [6] dataset, as reported in [14].

All three networks follow the encoder-decoder architecture paradigm. SegNet uses the convolutional layers of the VGG16 network [19] 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. The ENet architecture is based on a 'bottleneck' module, inspired by the residual blocks of ResNet [9]. Dilated convolutions are used in several bottleneck modules to increase the effective receptive field without losing resolution. ESPNet is also based on a module that exploits dilated convolutions. The ESP module uses a spatial pyramid of dilated convolutions to simultaneously learn multi-scale representations. Both ENet and ESPNet have smaller decoders than encoders, on the basis that the role of the decoder is primarily to upsample the low-resolution representation created by the encoder, fine-tuning the details, rather than learning new features.

4.2. Training

We train the networks on a subset of the ADE20k dataset [24], created by extracting images which contain maritime objects, as well as sea and sky. The ADE20k dataset 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. We manually exclude images which are completely unsuitable, e.g. an indoor scene which contains a painting of a boat, or images where sky is

Figure 3: Example training images from the subset of ADE20k [24].

only visible through a window. Examples of images in the subset for training can be seen in Fig. 3.

We map the original ADE20k classes to one of 4 classes (plus a void class): *Sea*, *Sky*, *Object* and *Other*. Note that the *Object* class refers to maritime objects, i.e. those that are found on the surface of the sea and that we want to detect in a maritime surveillance scenario. This includes large ships, speedboats, sailing boats, buoys, and so on. Any other objects are mapped to the *Other* class. The generated dataset consists of 448 images with median dimensions of 300×256 pixels. The properties of the data can be found in Table 2. Note that, on average, the target *Object* class occupies just 4% of the total image.

	Sea	*Sky*	*Object*	*Other*	*Void*
No. Image Occurrences	443	417	180	433	447
Total proportion (all images)	34%	30%	9%	25%	2%
Mean proportion (per image)	34.6%	32.9%	4%	26.6%	1.9%

Table 2: Training data properties.

We use the hyperparameters, class balancing scheme and training protocol as described in each of the original papers to train the networks (see Table 1). SegNet can be trained end-to-end; for ENet and ESPNet, the encoder and decoder are trained separately (using down-sampled groundtruth to train the encoder). For all networks, we use batches of 4 images, scaled to 640×480. During training, we apply random data augmentations (crops, horizontal flips, rotations, shears, and brightness/colour perturbations).

5. Experimental results

We train the networks as described above, using the author's original code where possible. SegNet[5] and ENet[6] are implemented in the Caffe framework; ESPNet[7] is implemented in the PyTorch framework. The proposed object detection system is implemented in Python, which has convenient interfaces to both frameworks. Training and object detection are run on the same 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.

5.1. Maritime surveillance datasets

We select sequences (Table 3) from 4 publicly available maritime surveillance datasets: Maritime Object Detection Dataset (MODD) [11], Singapore Maritime Dataset (SMD) [17], IPATCH [16] and SEAGULL [18]. These datasets span the range of maritime surveillance contexts, ranging from very low in the water (MODD) to high aerial (SEAGULL). We select sequences where object groundtruth (bounding boxes) is available, and which contain a range of technical challenges, including large camera motion, small/distant objects, glare/reflections, and wakes/whitepeaks. We also include a sequence where haze is present to compare performance under challenging visibility conditions. The sequences from the IPATCH dataset contain many frames where no objects are present, so we extract a shorter sub-sequence which contains a more balanced selection of frames with and without objects.

5.2. Evaluation metrics

To evaluate object detection performance, we adopt two widely-used detection metrics from the CLEAR 2006 evaluation [21]: N-MODA provides a score of detection accuracy for a whole sequence, taking into account missed detections and false positives, while N-MODP provides a corresponding sequence-level measure of detection precision (quality of object localisation). For N-MODA, we set $c_m = c_f = 1$. To complement these two scores, we also plot recall curves for differing overlap thresholds to measure how well each network can localise objects, regardless of false positives. Recall is computed as a function of overlap threshold, $\tau \in (0, 1]$:

$$\text{Recall}(\tau) = \frac{TP(\tau)}{N_G}, \ \ TP(\tau) = \sum_{t=1}^{N_{\text{frames}}} \sum_{i=1}^{N^t_{\text{matched}}} \left[\frac{|D^t_i \cap G^t_i|}{|D^t_i \cup G^t_i|} \geq \tau \right], \quad (1)$$

where N_G is the number of groundtruth targets in the whole sequence and $TP(\tau)$ is the number of true positives, calculated as the number of matched detections where the overlap ratio between estimated and groundtruth bounding boxes is at least τ. D^t_i and G^t_i represent the detected and groundtruth targets, respectively, for the i^{th} matching in frame t. Matching is performed using the spatial overlap ratio and the Hungarian algorithm, for consistency with the N-MODA and N-MODP metrics. As the networks perform classification and localisation jointly, we analyse the classification aspect by computing the metrics for two conditions: 1) where the best detection is taken, regardless of whether it is the correct class (*Object* or *Other*); 2) where only correctly classified objects are included in the performance evaluation (i.e. detections labelled as *Other* are removed).

[5] https://github.com/alexgkendall/SegNet-Tutorial
[6] https://github.com/TimoSaemann/ENet
[7] https://github.com/sacmehta/ESPNet

Dataset	Camera Height / Viewpoint	Sequence	Resolution	No. Frames	Objs. per Frame		Obj. Size Range (px)		Detection Challenges
				(*with objs.*)	Mean	Max	Width	Height	
MODD [11]	Very low Camera on USV[4]	01	640 × 480	540 (*540*)	3.2	5	6 - 202	5 - 267	LCM, OL
		07	640 × 464	641 (*589*)	1.5	2	11 - 244	20 - 212	LCM, OL
SMD [17]	Low; camera on speedboat	0797_VIS_OB	1920 × 1080	600 (*600*)	2.1	3	29 - 641	34 - 169	LCM, OL
	Medium Static camera on shore	1469_VIS	1920 × 1080	600 (*600*)	9.9	11	45 - 423	30 - 215	OL, DO
		1448_VIS_Haze	1920 × 1080	604 (*604*)	9.1	10	40 - 376	22 - 228	OL, DO, Haze
IPATCH [16]	Med-high Camera on large vessel	Sc2a_Tk1-CAM11	1920 × 1080	800 (*544*)	1.4	2	5 - 276	8 - 150	R, W
		Sc3_Tk2-CAM14	1920 × 1080	1026 (*1026*)	2.0	2	7 - 338	7 - 212	DO, R, W
SEAGULL [18]	Very high Camera in aerial vehicle	lanchaArgos_clip3	1920 × 1080	1401 (*1238*)	1.0	1	4 - 216	8 - 84	DO, R
		portimao_GP020175_part01	1920 × 1080	300 (*264*)	2.0	2	5 - 69	6 - 34	DO, R

Table 3: Maritime surveillance sequences. Key – LCM: large camera motion, OL: overlapping objects, DO: distant objects, R: reflections, W: wake.

5.3. Results and analysis

Results for the N-MODA and N-MODP scores are listed in Table 4. Qualitative examples of the segmentations produced by each network are presented in Fig 5. When class is ignored, ENet has the best detection accuracy (N-MODA score) overall; however, this performance degrades severely when class is taken into consideration. Although ENet detects objects of interest with few false positives, it tends to misclassify them as *Other* (e.g. Fig. 5 (f-h)). SegNet and ESPNet have lower detection accuracy, but are better at detecting *Objects* specifically, reflected in the higher *Object only* N-MODA scores. ESPNet suffers from a large number of false positives (of both classes) (e.g. Fig. 5 (j-l)), so achieves lower N-MODA scores. Looking at precision (N-MODP), ENet and SegNet perform best across all sequences when class is ignored, with ENet achieving a slightly higher average score. As before, this is largely due to the high false positive rate of ESPNet, which generates larger (and therefore less precise) regions (e.g. Fig. 5 (j)). However, in the SEAGULL-portimao sequence, both SegNet and ENet are unable to detect the target at all, whereas ESPNet can. This property of ESPNet becomes even more obvious when looking at the N-MODP scores with class labels taken into account. ESPNet is better at distinguishing maritime objects from other classes, despite its high false positive rate.

The recall curves (Fig. 4) allow a more fine-grained analysis of the trade off between localisation accuracy and missed detections (false negatives), which is useful for deciding which network to use in a particular application. For example, if the cost of a missed detection is high, a greater number of false positives may be tolerated, especially if they can be filtered out using a secondary processing step. ENet and SegNet have comparable performance in most sequences, but ESPNet is better at outputting the correct class, as shown by the small difference in its recall curves (solid and dotted lines), compared to SegNet and ENet.

The generalisation of SegNet and ENet from the training data to the test sequences is promising, considering the small quantity of training data and the large difference in resolution and visual characteristics between the training and test sets. The poor visibility in SMD-1448_Haze proved more challenging for ENet, but SegNet maintained its recall performance under these conditions. Wakes and glare/reflections remain challenging for all networks. The reflection of the sun in the SEAGULL-lanchaArgos sequence proved particularly challenging (see Fig. 5 (d, h & l)) and was consistently detected as *Object* or *Other*. We speculate that this is due to the mismatch between the training data – which consists primarily of carefully framed photographs which minimise such artefacts – and the test sequences. *Objects* (boats, etc.) in the training data were also often white or light in colour.

A fundamental weakness in using semantic segmentation for object detection is that overlapping objects of the same class cannot be distinguished. This occurred in the MOD and SMD sequences, and influenced the scores for all networks, compared to sequences which contain only 1 or 2 non-overlapping objects. This issue could be addressed by combining the network output with an edge or feature detector to infer boundaries between objects. Finally, an interesting 'by-product' of the proposed approach is that the horizon line can be easily inferred from the segmentation map, as has been done recently [10]. Horizon detection is a common pre-cursor task in maritime surveillance, as it can be used to determine camera orientation and inferring the distance (and hence real-world size) of objects.

6. Conclusion

In this paper, we have evaluated the performance of deep semantic segmentation networks for object detection in maritime surveillance applications and demonstrated the feasibility of the approach. Even with very limited training data, the ability to generalise to different viewpoints, environmental conditions and object types is promising. Overall, SegNet and ENet achieve higher detection accuracy and precision, but ESPNet is better at classifying objects correctly. Considering the maritime surveillance application, the ENet model would be the most suitable from this study,

[4] Unmanned Surface Vehicle

Sequence	N-MODA						N-MODP					
	Object and *Other*			*Object* only			*Object* and *Other*			*Object* only		
	SegNet	ENet	ESPNet	SegNet	ENet	ESPNet	SegNet	ENet	ESPNet	SegNet	ENet	ESPNet
MODD-01	**0.676**	0.200	0.397	0.398	**0.457**	0.453	**0.150**	0.128	**0.150**	0.150	0.109	**0.170**
MODD-07	0.378	**0.683**	0.003	0.506	0.159	**0.626**	**0.338**	0.315	0.292	0.088	0.009	**0.270**
SMD-0797_VIS_OB	**0.447**	0.256	-0.295	**0.501**	0.343	0.276	0.273	0.496	**0.524**	**0.212**	0.046	0.010
SMD-1469_VIS	0.167	**0.644**	0.186	**0.859**	0.142	0.755	**0.334**	0.316	0.131	0.058	0.016	**0.112**
SMD-1448_VIS_Haze	0.404	**0.494**	0.423	0.166	0.090	**0.679**	0.171	**0.210**	0.088	**0.147**	0.005	0.040
IPATCH-Sc2a_Tk1-CAM11	-3.471	**0.382**	-11.320	**0.388**	0.225	-3.858	0.198	**0.211**	0.136	0.010	0.002	**0.104**
IPATCH-Sc3_Tk2-CAM14	**-3.796**	-3.912	-8.322	**0.354**	-0.767	-3.022	0.265	**0.355**	0.143	0.004	0.028	**0.129**
SEAGULL-lanchaArgos	-8.501	**-5.656**	-6.254	**-1.319**	-1.686	-2.092	**0.432**	0.420	0.168	0.050	0.001	**0.168**
SEAGULL-portimao	0.013	0.048	**0.491**	0.013	0.000	**0.565**	0.000	0.000	**0.242**	0.000	0.000	**0.242**
Mean	-1.520	**-0.762**	-2.743	**0.207**	-0.115	-0.624	0.240	**0.272**	0.208	0.080	0.024	**0.138**

Table 4: N-MODA and N-MODP scores for two test conditions: 1) class is ignored (both *Object* and *Other* detections are included in the evaluation); 2) *Other* detections are discarded before evaluation.

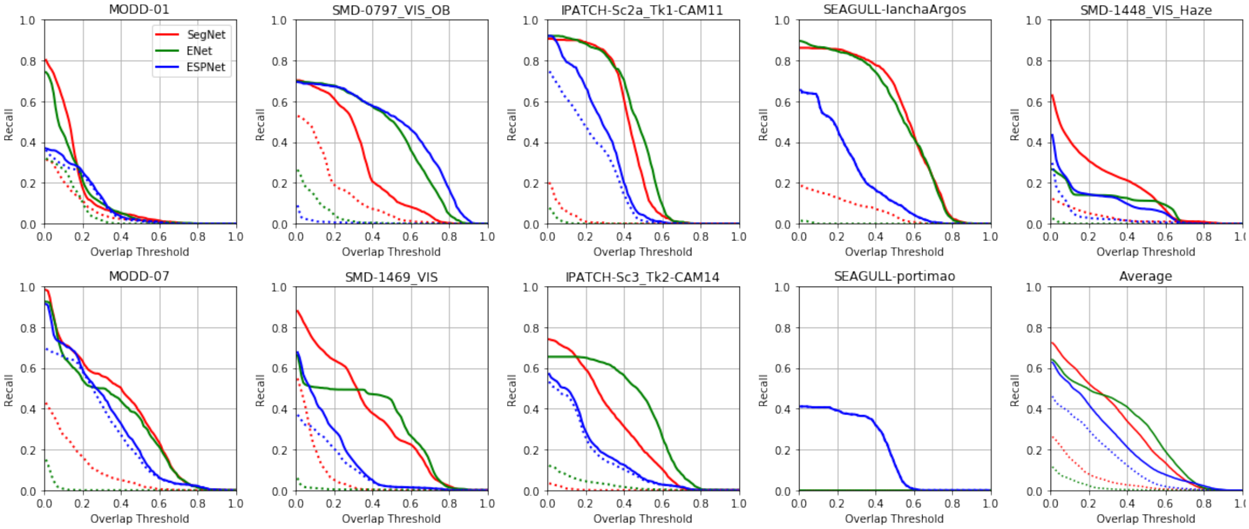


Figure 4: Recall curves for varying overlap threshold τ. Solid line represents test condition 1 (class ignored); dotted line represents test condition 2 (*Other* detections discarded before evaluation).

as it is faster than SegNet. Future work will explore using additional training data that is more representative of maritime surveillance sequences to improve performance, and investigate whether the proposed approach could be combined with other methods to improve separation of overlapping objects.

References

[1] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *Trans. PAMI*, 39(12):2481–2495, 2017.

[2] D. D. Bloisi, F. Previtali, A. Pennisi, D. Nardi, and M. Fiorini. Enhancing Automatic Maritime Surveillance SystemsWith Systems With Visual Information. *Trans. ITS*, 18(4):824 –833, 2017.

[3] F. Bousetouane and B. Morris. Fast CNN surveillance pipeline for fine-grained vessel classification and detection in maritime scenarios. In *AVSS*, pages 242–248, 2016.

[4] G. J. Brostow, J. Fauqueur, and R. Cipolla. Semantic object classes in video: A high-definition ground truth database. *Patt. Recog. Lett.*, 30(2):88–97, 2009.

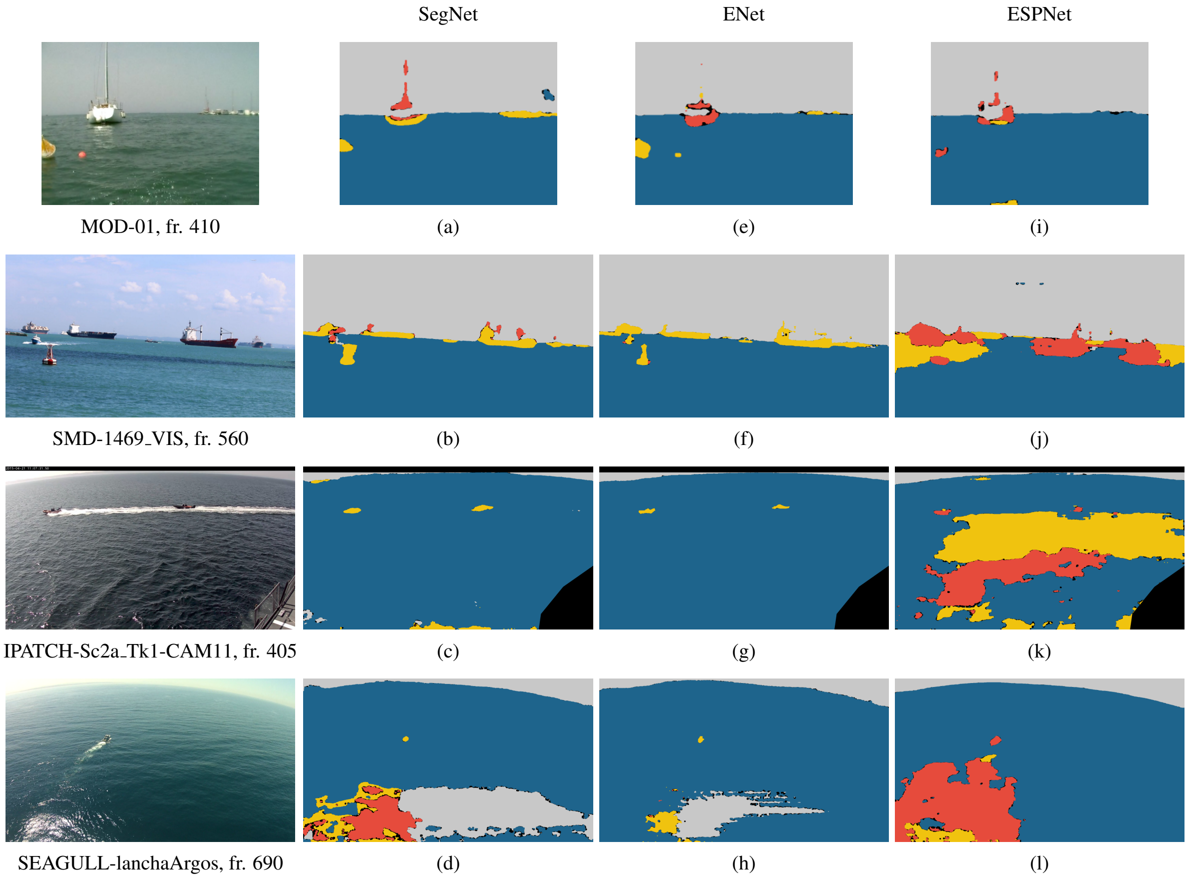
[5] T. Cane and J. Ferryman. Saliency-Based Detection for Maritime Object Tracking. In *CVPR Work.*, pages 1257–1264, 2016.

[6] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The Cityscapes Dataset for Semantic Urban Scene Understanding. In *CVPR*, 2016.

[7] G. Cruz and A. Bernardino. Aerial Detection in Maritime Scenarios Using Convolutional Neural Networks. In *ACIVS*, pages 373–384, 2016.

[8] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, and J. Garcia-Rodriguez. A Review on Deep Learning Techniques Applied to Semantic Segmentation. *arXiv:1704.06857*, 2017.

[9] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. In *CVPR*, pages 770–778, 2016.



SegNet

ENet

ESPNet

MOD-01, fr. 410

(a)

(e)

(i)

SMD-1469_VIS, fr. 560

(b)

(f)

(j)

IPATCH-Sc2a_Tk1-CAM11, fr. 405

(c)

(g)

(k)

SEAGULL-lanchaArgos, fr. 690

(d)

(h)

(l)

Figure 5: Example pixel classifications from each network (after temporal smoothing and morphological filtering) for representative frames from each dataset: (a-d) SegNet [1], (e-h) ENet [15], (i-l) ESPNet [14]. Key: Blue = *Sea*, Grey = *Sky*, Red = *Object*, Yellow = *Other*, Black = masking provided for IPATCH sequence which we apply to the output class probabilities of each network before processing.

[10] C. Y. Jeong, H. Yang, and K.-D. Moon. Horizon detection in maritime images using scene parsing network. *IET Electron. Lett.*, 2018.

[11] M. Kristan, V. S. Kenk, S. Kovačič, and J. Perš. Fast Image-Based Obstacle Detection from Unmanned Surface Vehicles. *Trans. Cybern.*, 46(3):641–654, 2016.

[12] M. J. Loomans, R. G. Wijnhoven, and P. H. de With. Robust automatic ship tracking in harbours using active cameras. In *ICIP*, pages 4117–4121, 2013.

[13] J. Marques, A. Bernardino, G. Cruz, and M. Bento. An Algorithm for the Detection of Vessels in Aerial Images. In *AVSS*, pages 295–300, 2014.

[14] S. Mehta, M. Rastegari, A. Caspi, L. Shapiro, and H. Hajishirzi. ESPNet: Efficient Spatial Pyramid of Dilated Convolutions for Semantic Segmentation. *arXiv:1803.06815*, 2018.

[15] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello. ENet: A Deep Neural Network Architecture for Real-Time Semantic Segmentation. *arXiv:1606.02147*, 2016.

[16] L. Patino, T. Nawaz, T. Cane, and J. Ferryman. PETS 2017: Dataset and Challenge. In *CVPR Work.*, pages 2126–2132, 2017.

[17] D. K. Prasad, D. Rajan, L. Rachmawati, E. Rajabally, and C. Quek. Video Processing From Electro-Optical Sensors for Object Detection and Tracking in a Maritime Environment : A Survey. *Trans. ITS*, pages 1–24, 2017.

[18] Ricardo Ribeiro. The SEAGULL dataset, http://vislab.isr.ist.utl.pt/seagull-dataset.

[19] K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv:1409.1556*, pages 1–14, 2014.

[20] A. Sobral, T. Bouwmans, and E. H. Zahzah. Double-constrained RPCA based on saliency maps for foreground detection in automated maritime surveillance. In *AVSS*, 2015.

[21] R. Stiefelhagen, K. Bernardin, R. Bowers, J. Garofolo, D. Mostefa, and P. Soundararajan. The CLEAR 2006 Evaluation. In *CLEAR Work.*, pages 1–44, 2006.

[22] T. H. Tran and T. L. Le. Vision based boat detection for maritime surveillance. In *ICEIC*, 2016.

[23] Y. Zhang, Q. Z. Li, and F. N. Zang. Ship detection for visual maritime surveillance from non-stationary platforms. *Ocean Eng.*, 141:53–63, 2017.

[24] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba. Scene parsing through ADE20K dataset. In *CVPR*, pages 5122–5130, 2017.