

# Deep Visual Place Recognition for Shoreline Navigation

Luke Thomas

Submitted to Swansea University in fulfilment  
of the requirements for the Degree of Doctor of Philosophy



**Swansea University**  
**Prifysgol Abertawe**

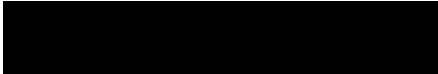
Department of Computer Science  
Swansea University

May 11, 2025

Copyright: The author, Luke Thomas, 2025

# Declaration

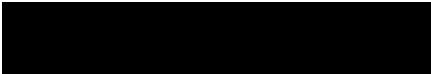
This work has not been previously accepted in substance for any degree and is not being concurrently submitted in candidature for any degree.

Signed  (candidate)

Date 10/06/2025.....

# Statement 1

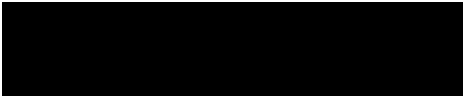
This thesis is the result of my own investigations, except where otherwise stated. Other sources are acknowledged by footnotes giving explicit references. A bibliography is appended.

Signed  (candidate)

Date 10/06/2025.....

# Statement 2

I hereby give my consent for my thesis, if accepted, to be available for photocopying and for inter-library loan, and for the title and summary to be made available to outside organisations.

Signed  (candidate)

Date 10/06/2025.....



# Abstract

The use of visual place recognition (VPR) to identify approximate geographical locations from land-based imagery has seen great success in recent history, generally methods of performing VPR work by converting images into a set of representative feature descriptors which can then be compared mathematically to determine a closest match. Originally, these descriptors were generated using hand-crafted methods such as Scale Invariant Feature Transform (SIFT), however in more recent years we have seen the development of Deep VPR, extending the methodology to use outputs from Convolutional Neural Networks (CNN) as the basis for feature descriptors instead, an idea that has led to the development of multiple state-of-the-art methods in the last decade. However, almost all of these works are focused on land-based imagery exclusively which makes sense given that one of the biggest motivators behind the development of Deep VPR is assisting in the navigation of autonomous cars and robots. Our work seeks to test the viability of Deep VPR for Shoreline Imagery for the purpose of sea vessel navigation, where images still contain nearby land features from the visible shoreline but are taken from a vessel out on the water thus introducing a drastically different perspective from the types of images most state-of-the-art Deep VPR models have been trained and evaluated on.

In this thesis we provide a new in-house dataset containing images generated from several recordings of travels across the Plymouth Sound, UK over multiple days during March and April 2022 provided by a mounted camera system placed on the IBM/Promare Mayflower Autonomous Ship. This dataset forms our benchmark for evaluating Deep VPR performance on Shoreline Imagery.

We first show a set of results and insights on our initial application of Deep VPR to shoreline imagery and compare these to traditional land-based locational imagery. Using a novel image salience technique to highlight what specific key features in each of the two categories our CNN architecture is picking up on. As this work was carried out during COVID our in-house dataset could not be generated at the time and as such the Symphony Lake dataset whose

images are somewhat shoreline-adjacent is used as a stand-in.

Secondly, with our in-house dataset then available, we carry out a series of experiments based around the modification of a state-of-the-art Deep VPR pipeline in order to exploit salient feature regions in Shoreline Imagery, as well as tackle the issue of feature redundancy. Our experiments lead us to a novel domain-specific pipeline that provides new state-of-the-art results on our in-house dataset.

This pipeline is then used as the basis for a novel human-centered study analysing trust in Deep VPR for Shoreline imagery; the study is made up of several independent surveys including a control survey that simply shows a series of image matching results from the pipeline, a second survey making use of our previously discussed saliency visualisations to communicate model feature extraction to the user explicitly, and surveys that allow the user to intervene in the models decision making directly.

The outcome of this work is to show how Deep VPR translates to the shoreline image domain, how the features of this domain differ from land-based imagery, and how we can build pipelines to take advantage of these differences to achieve better results, and how we can ensure user trust in these Deep VPR pipelines under real-life navigation scenarios using various human-computer interaction techniques.

# Acknowledgements

I would like to give thanks to Dr Mike Edwards who supervised for the majority of my studies, who has always been available to provide advice, critical discussion and direction for this thesis for which I am incredibly thankful for. In addition I would like to thank him for always bringing a calm and confident mindset to our discussions that I believe has influenced my own and helped me to develop as a person.

In addition, I would like to thank Dr Matt Roach who took on the role of supervisor later into this thesis, providing the idea for and overseeing the Deep VPR user trust studies. This chapter was a big shift into HCI for the project and myself and Matt Roach helped to make that shift much smoother than it otherwise would have been.

As well as my supervisors, I have always been surrounded by the incredibly bright minds of the EPSRC CDT at Swansea University who have always been open to the exchanging of ideas, concepts and solutions to various problems. In particular I wish to extend my gratitude to Connor Clarkson, who has always been available for thought provoking discussion.

Thanks to the team at UK Hydrographics Office for providing additional insight into the project, in particular Mark Casey who put out the initial request for this research to be undertaken and Austin Capsey who acted as an additional secondary supervisor and our main point of contact for the UKHO.

For the in-house dataset my deepest thanks go to the team at Marine AI Plymouth who worked on the IBM/Promare Mayflower Autonomous Ship, as without the dataset much of this work would have not been possible.

# Contents

<b>List of Tables</b>	<b>11</b>
<b>List of Figures</b>	<b>12</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Introduction to Problem . . . . .	2
1.2 Motivations . . . . .	3
1.3 Objectives . . . . .	4
1.4 Overview . . . . .	5
1.5 Contributions . . . . .	6
1.6 Publications . . . . .	8
1.7 Outline . . . . .	9
<b>2 Background of Visual Place Recognition</b>	<b>11</b>
2.1 Introduction . . . . .	13
2.2 Local Descriptor-Based Visual Place Recognition . . . . .	14
2.2.1 Local Descriptor Extraction . . . . .	14
2.2.2 Local Descriptor Matching . . . . .	17
2.3 Global Descriptor-Based Visual Place Recognition . . . . .	19
2.3.1 Global Descriptor Extraction . . . . .	19
2.3.2 Global Descriptor Matching . . . . .	20
2.4 Feature Descriptor Robustness to Perspective and Distance . . . . .	22
2.5 Dimensionality Reduction . . . . .	24
2.5.1 General-purpose Methods . . . . .	24
2.5.2 Image-based Methods for VPR . . . . .	26
2.6 Retrieval Refinement . . . . .	27

2.7	Evaluating Visual Place Recognition Models . . . . .	28
2.7.1	Definition of Ground Truth . . . . .	28
2.7.2	Metrics for Evaluation . . . . .	29
2.8	Methods for Real-time Localization . . . . .	30
2.8.1	Simultaneous Localization and Mapping . . . . .	31
2.8.2	Structure from Motion . . . . .	32
2.8.3	Multi-Sensor Fusion . . . . .	33
2.9	User Reliance on AI for Autonomous Navigation . . . . .	34
2.9.1	Background . . . . .	34
2.9.2	Evaluating User Reliance . . . . .	35
<b>3</b>	<b>Computer Vision Task Backgrounds and Applications for Waterborne Imagery</b>	<b>38</b>
3.1	Introduction . . . . .	40
3.2	Object Detection/Region Proposal . . . . .	41
3.2.1	Handcrafted Methods . . . . .	42
3.2.2	Deep Learning Methods . . . . .	45
3.2.3	Applications in Waterborne Imagery . . . . .	48
3.3	Image Saliency . . . . .	50
3.3.1	Handcrafted Methods . . . . .	51
3.3.2	Deep Learning Methods . . . . .	53
3.3.3	Applications in Waterborne Imagery . . . . .	59
3.4	Semantic Segmentation . . . . .	62
3.4.1	Fully Convolutional Networks . . . . .	63
3.4.2	Deconvolutional Networks . . . . .	65
3.4.3	ResNet-based Fully Convolutional Networks . . . . .	68
3.4.4	Applications in Waterborne Imagery . . . . .	71
3.5	Human-Centered AI . . . . .	72
3.5.1	Explainable AI . . . . .	73
3.5.2	Human-in-the-Loop . . . . .	75
<b>4</b>	<b>Creating the Plymouth Sound Dataset</b>	<b>77</b>
4.1	Data Collection . . . . .	78
4.1.1	Image Collection . . . . .	78
4.1.2	Locational Information Collection . . . . .	80

4.2	Dataset Features . . . . .	81
4.2.1	Perspective Changes . . . . .	83
4.2.2	Appearance Changes . . . . .	83
4.3	Availability . . . . .	84
4.4	Challenges . . . . .	85
4.4.1	Overabundance of Non-discriminative Features . . . . .	85
4.4.2	Obstructions . . . . .	87
<b>5</b>	<b>Evaluating Waterborne Deep VPR</b>	<b>88</b>
5.1	Introduction . . . . .	89
5.2	Proposed Approach . . . . .	90
5.3	Method . . . . .	90
5.3.1	Datasets . . . . .	91
5.3.2	Architecture . . . . .	93
5.4	Comparative Analysis and Results . . . . .	96
5.4.1	Quantitative Analysis . . . . .	98
5.4.2	Qualitative Analysis . . . . .	99
5.5	Summary . . . . .	100
<b>6</b>	<b>Improvement of Waterborne Deep VPR</b>	<b>103</b>
6.1	Introduction . . . . .	104
6.2	Proposed Approach . . . . .	105
6.3	Datasets . . . . .	106
6.3.1	Plymouth Sound Dataset . . . . .	106
6.3.2	Symphony Lake Dataset . . . . .	106
6.4	Architectures . . . . .	106
6.4.1	SSM-VPR (Baseline) . . . . .	106
6.4.2	SSM-VPR w/ Stage 1 Selective Search Region Proposal . . . . .	107
6.4.3	SSM-VPR w/ Stage 1 rOSD Region Proposal . . . . .	108
6.4.4	SSM-VPR w/ Stage 2 WaSR Semantic Line-based Region Proposal (SHM-VPR) . . . . .	108
6.5	Comparative Analysis and Results . . . . .	111
6.5.1	Quantitative Analysis . . . . .	111
6.5.2	Qualitative Analysis . . . . .	114

6.6	Summary . . . . .	120
<b>7</b>	<b>Semantic Segmentation based Knowledge priors for Waterborne Deep VPR</b>	<b>121</b>
7.1	Introduction . . . . .	122
7.2	Proposed Approach . . . . .	123
7.3	Dataset . . . . .	123
7.4	Architectures . . . . .	124
7.4.1	SSM-VPR (i.e. Baseline) . . . . .	124
7.4.2	SSM-VPR w/Segmentation Enhanced Feature Map (SEFM) [1] . . . . .	125
7.4.3	SSM-VPR w/Semantically Aware Local Descriptor Refinement (SALDR) [2] . . . . .	125
7.4.4	SHM-VPR Semantic Edge based SSM-VPR Stage 2 Spatial Descriptor Matching . . . . .	127
7.4.5	Semantically Aware SSM-VPR . . . . .	129
7.5	Comparative Analysis and Results . . . . .	129
7.5.1	Quantitative Analysis . . . . .	129
7.5.2	Qualitative Analysis . . . . .	133
7.6	Summary . . . . .	137
<b>8</b>	<b>Measuring User Reliance and Technology Dominance in Waterborne Deep VPR With Human-Centered AI</b>	<b>139</b>
8.1	Introduction . . . . .	140
8.2	Method . . . . .	141
8.2.1	Architectures . . . . .	141
8.2.2	Dataset . . . . .	143
8.2.3	User Study . . . . .	144
8.3	Comparative Analysis and Results . . . . .	151
8.3.1	Quantitative Analysis . . . . .	151
8.3.2	Qualitative Analysis . . . . .	156
8.4	Conclusions . . . . .	159
<b>9</b>	<b>Conclusions and Future Work</b>	<b>161</b>
9.1	Conclusions . . . . .	161
9.2	Contributions . . . . .	164

9.3 Future Work . . . . . 165

**Bibliography** . . . . . **168**



# List of Tables

4.1	GPGGA Data Format . . . . .	81
4.2	GNVTG Data Format . . . . .	82
4.3	GNHDT Data Format . . . . .	82
4.4	GPZDA Data Format . . . . .	82

# List of Figures

2.1	Left, the octave methodology used by SIFT to obtain numerous scale space images with neighbouring pairs subtracted to produce DoG. Right, SIFT finds local extrema via pixels with maximum and minimum values against both immediate neighbours and those from the above and below DoG's. Figures taken from [3]. . . . .	15
2.2	Depiction of how SURF box filter scaling differs from SIFT image scaling, on the left SIFT uses subsampling to analyse the image at different scales using DoG whereas on the right SURF changes the scale of it's Fast-Hessian detector box [4] . . . . .	16
2.3	The SSM-VPR [5] pipeline which extracts sub-region based local vectors from CNN image feature maps in a two-stage approach. . . . .	17
2.4	Figure depicting the NetVLAD pipeline, given an image a backbone CNN is used to output a feature map which is the used as input for the NetVLAD layer itself which soft-assigns the features to several clusters. Figure taken from [6]. . . . .	20
2.5	Generation of a confusion matrix for visual place recognition, features are extracted from each test image via CNN backbone and matched to those from all training images. Each matrix element $M(i, j)$ represents the Euclidean distance between the $i^{th}$ training image and the $j^{th}$ testing image. Figure taken from [7]. . . . .	21
2.6	The boxes in the figure above show landmark proposals extracted by EdgeBoxes that have been converted into convolutional feature descriptors for image matching. From both examples, you can see that the two image pairs were able to be matched from significantly different viewpoints. . . . .	23
2.7	Multiple instances of the same building at different perspectives and distances matched to the query using the R-MAC representation. . . . .	23
2.8	The SLAM problem as defined in [8], both the robot and landmark positions are mapped and localized simultaneously. . . . .	31
2.9	Results of structuring the coliseum in Rome using SfM [9]. . . . .	32

2.10	Results of grouping words ranging from high average distrust association to high average trust association into related phrases [10] . . . . .	36
3.1	Examples of pedestrian detection training images showing the large variety of poses, illumination and clutter that one could encounter when trying to perform detection. Figure taken from [11]. . . . .	42
3.2	Examples of rectangular features used by Viola-Jones method [12] . . . . .	43
3.3	Visual depiction of the Selective Search exhaustive segmentation followed by several merges over the iterations, below each segmentation image are the resulting bounding-boxes [13]. . . . .	44
3.4	The R-CNN pipeline [14], which uses Selective Search for region proposals that are then warped and passed through a CNN for individual object classification. . .	45
3.5	Fast-RCNN pipeline [15], which does away with the image warping stage and instead projects RoI's onto the CNN feature map, shape consistency is then ensured via the RoI pooling layer. . . . .	46
3.6	Faster-RCNN pipeline [16], using a trained RPN to generate proposals based on feature maps which are RoI pooled and classified. . . . .	46
3.7	The Vo et al. method for Object Detection/Region Proposal. The top row depicts, from left to right, an input image, a summed CNN feature map, local maxima extracted from the map and three individual feature maps generated by calculating the dot product between the CNN feature vector at the position of a maxima and the feature map. Subsequent rows then depict these same three feature maps, followed by examples at different saliency thresholds of the main connected component generated from the maps being used to generate bounding boxes. . . . .	47
3.8	Images taken from Figures in Bloisi et al.'s work [17], left image shows how existing object detectors cannot distinguish portions of land from boats and right image shows a low-resolution image of a boat with significant wake behind, which has been falsely detected as a second boat. . . . .	48
3.9	Horizon line detection method from Bloisi et al. [17], candidate lines are extracted from the input image (a) by applying a hough transform to its edge map, producing image (b). Candidates are validated by taking a rectangular set of sample points above and below the line, where corresponding pairs above and below are checked for differing intensities (See Image (c)), if 90% of pairs differ the line is considered valid. . . . .	49

3.10	The Copy-and-Paste and Mix-up technique pipeline used by Kime et al. [18] on images from the Singapore Maritime Detection (SMD) dataset. These image augmentations are used to more effectively train YOLO-V5 on the SMD data. . . . .	50
3.11	Itti model general architecture [19] . . . . .	52
3.12	Li et al. [20] architecture for visual saliency detection using multiscale deep CNN features. . . . .	54
3.13	BASNet [21] architecture for boundary-aware salient object detection. . . . .	55
3.14	Taken from the Mask paper [22], we can see that by blurring the flute object within the image (i.e. Perturbing the image), we can reduce it's class score for flute from 0.9973 to 0.0007, from which we can learn a "mask" which acts as a saliency map indicating the importance of the object to classification. . . . .	56
3.15	RISE [23] architecture for generating explainable saliency via linear combinations of loss based on a list of masked perturbations of the image input. . . . .	57
3.16	CAM saliency [24]. Global average pooling weights are mapped back and multiplied by their activation maps, which are linearly combined into the saliency map. . . . .	58
3.17	The Grad-CAM [25] architecture, here the desired output of the classification task is backpropogated to the rectified convolutional feature maps of interest, which get combined to produce the Grad-CAM heatmap. As seen in the figure, Grad-CAM also supports a guided variant by multiplying the heatmap with the networks guided backpropagation. . . . .	58
3.18	Score-CAM architecture from Wang et al. [26]. Initial activation maps are up-sampled to produce $M_i$ masks for an image $I$ , after which it essentially mimics the RISE architecture, multiplying the masks by $I$ to produce $I \odot M_i$ , each produces a score which can be mapped back to the activation maps and these can be summed to produce saliency. . . . .	59
3.19	The Visual Maritime Attention framework [27], which uses associated edge ( $E$ ), right angle ( $RA$ ), high frequency ( $HF$ ), contrast ( $C$ ) and colour ( $CL$ ) metrics to calculate density ( $D$ ), dissimilarity ( $X$ ) and surround features ( $S$ ). Additionally, a sea and sky detector which takes in a HSV-Colour version of the image input is used, producing an 18-channel histogram with $20^\circ$ separation, which is trained via Naive Bayes Classifier. All four features are aggregated and Naive Bayes Classification is used to get the final mapping. . . . .	60

3.20	By using an initial salience map to inform the generation of both a confidence map and shape constraint, RPCA can be applied on top of these to produce a more refined segmentation for maritime imagery [28]. . . . .	60
3.21	From left to right: (a) shows three input images, (b) shows BMS saliency map [29] (c) shows Ground-truth and (d) shows the RPCA method [28]. . . . .	61
3.22	Grad-CAM [25] outputs for a CNN classifier trained on the Wright and Logan photographic collection within the National Museum of the Royal Navy [30]. In the top left, we see that Grad-CAM successfully highlights the sail of the submarine, top right the aircraft carriers ramp is highlighted, bottom left the minesweepers specialized equipment is highlighted and the bottom right describes a large edge detection filter around the large dreadnought. . . . .	62
3.23	The FCN [31] uses convolutional output layers to produce a dense pixel-wise prediction map in a supervised end-to-end architecture. . . . .	63
3.24	The U-Net architecture [32] uses a symmetrical contractive/expansive fully convolutional model in order to produce high-level semantic feature maps that can then be gradually upsampled into classified spatial features. Between corresponding convolutions on either side are skip connections, which re-introduce the semantic knowledge gained from the contractive stage to the expansive one. . . . .	64
3.25	The Parallel FCN architecture from [33], the object segmentation branch is equivalent to a normal FCN, the edge extraction branch uses multiple VGG convolutional side outputs to fuse into a edge detection map. Outputs from these branches are then passed through a domain transform for a more refined output . . . . .	65
3.26	DeconvNet [32] makes use of a standard VGG-based CNN to output a dense prediction output which is convolved and unpooled gradually to produce a semantic segmentation map. . . . .	65
3.27	An illustration of the two contextual modules employed by CDN [34], (A) Channel Contextual Module and (B) Spatial Contextual Module . . . . .	67
3.28	A residual block that takes an initial input $x$ , passes it through various layers to get an output $F(x)$ which is then aggregated to the original $x$ via the skip connection (noted here as ‘identity’). . . . .	69

3.29	The PSPNet [35] architecture, which depicts the pyramid pooling module where outputs from ResNet CNN are passed through four pooling layers of increasing feature map size. These feature maps are convolved with a 1x1 filter, upsampled and appended to the CNN output. . . . .	69
3.30	The MP-ResNet architecture [36], uses a standard FCN architecture with ResNet encoding, after encoding 2, 2 additional branches form applying ResNet modules 3 and 4 to the encoding 2, 3 and 4 in different orders in parallel. These outputs are then fused in the decoding phase. . . . .	70
3.31	WaSR architecture [37], encoder generates high-level feature maps that are fused with the decoder, optional IMU feature channel provides assistance in detecting the visible water edge for more accurate segmentation. . . . .	72
3.32	Each row depicts an input image and multiple segmentation maps output by several architectures for land (yellow), sea (light blue) and sky (dark blue). From left to right we have an input image, SegNet output, RefineNet output, <i>WaSR<sub>WSL</sub></i> and <i>WaSR<sub>WSSL</sub></i> . In columns 2 and 3 there are visible errors within the outputs highlighting sky as sea or sea as land pixels etc. due to wakes, reflections and weather conditions. Columns 4 and 5 show two sub-variants of WaSR, each of which is more resistant to these challenging conditions. . . . .	73
3.33	Figure taken from [38] depicting the two branches of Explainable AI at work. Boxes labelled a). b). and c). mark three ways in which a Neural Network classifier identifies an input as a “horse”. Box a). depicts a graph showing highly activated neurons that lead to the output decision which are mapped to the analysis in box b). indicating features that are often activated by these neurons to give the developer knowledge through Transparency Design. Box c). then shows how this information can be used to produce a Post-Hoc Explanation for the user. . . . .	74
4.1	Left: Image of the IBM/Promare Mayflower Autonomous Ship (MAS) which was used for image capture with its multi-view camera system mounted. Right: Simple Diagram of the arrangement of the multi-view camera system, each circle represents a single camera and ID. . . . .	79
4.2	Maps depicting the path of each run along the Plymouth Sound and beyond. . . . .	79
4.3	An image set from MAS’s multi-view camera system. Top row (left to right): Images from camera ID 0, 1 and 2. Bottom row (left to right): Images from camera ID 3, 4 and 5 . . . . .	83

4.4	Top Row: Images of the Rame’s Head taken from run 1 during the MAS’s embarkment (left) versus disembarkment (right). Bottom Row: An additional embarkment/disembarkment image pair of the Rame’s Head from run 4. . . . .	84
4.5	Similar views of Fort Picklecombe from runs 1-6, between these there are small differences in hue caused by cloud coverage as well as greater differences caused by fog such as in run 3 (Top Right). . . . .	84
4.6	Top Row: Images from Run 2 (03-31-2022) which were taken during exceptionally clear weather. Bottom Row: Images from Run 7 (04-14-2022) which were taken during heavy fog. . . . .	85
4.7	The average % of pixels within Plymouth Sound Dataset images belonging to the land, sea and sky categories defined by WaSR semantic segmentation. . . . .	86
4.8	Top Row: Images facing the bow of the MAS which acts as a distractor. Middle Row: Images facing the stern containing the observation boat accompanying the MAS. Bottom Row: Images from run 3 which had the camera obstructed by water. . . . .	87
5.1	Examples of ground truth image pairs of the same within the three Berlin image sets. Top Row: Halenseestrasse. Middle Row: Kudamm. Bottom row: A100. . . . .	91
5.2	Example Images from Symhpony Lake dataset. From Top Row to Bottom: Images taken from 2014, 2015, 2016 and 2017. . . . .	92
5.3	A diagram of the overall pipeline of SSM-VPR, including Stage 1: Image Filtering and Stage 2: Spatial Matching. . . . .	93
5.4	Figure inspired by the original paper [5]. A simplified representation of the spatial matching stage for a grid of query and retrieval vectors, taking a pair of anchor points between the two, their surrounding vectors along the row and column should also match if the features are spatially consistent. . . . .	95
5.5	Score-CAM architecture from Wang et al. [26]. . . . .	95
5.6	A simple diagram of Score-CAM adaptation for SSM-VPR. Top Row: A Query and Retrieval Image pair. Second Row: Retrieval Image is passed through CNN to get feature map filters. Third Row: The Retrieval is multiplied by these filters to produce masked versions, these are passed through the Deep VPR pipeline to obtain vectors for each masked image. Bottom Row: A distance metric is calculated between each masked vector and the queries vectors and used to weight the masks, the linear combination produces the saliency map. . . . .	97

5.7	Left: PR-curves for Berlin and Symphony lake test folds. Right: Corresponding Precision at Top K curves. . . . .	98
5.8	3x4 Table depicting a series of queries and associated image retrievals row-by-row from the Berlin Halenseestrasse image set. First Column: Image Queries. Second-Fourth Column: 1st-3rd highest scoring retrievals for each query. . . . .	100
5.9	3x3 Table depicting Query Images (First Column), their rank 1 retrievals (Second Column) and the result of applying Score-CAM to the rank 1 retrievals (Third Column). . . . .	101
5.10	3x4 Table depicting a series of queries and associated image retrievals row-by-row from the Symphony Lake image set. First Column: Image Queries. Second-Fourth Column: 1st-3rd highest scoring retrievals for each query. . . . .	102
5.11	3x3 Table depicting Query Images (First Column), their rank 1 retrievals (Second Column) and the result of applying Score-CAM to the rank 1 retrievals (Third Column). . . . .	102
6.1	Diagram of the proposed SSM-VPR w/ Stage 1 Selective Search Region Proposal, which takes a set of region proposals from selective search based on a 2D pseudo-image and uses them to select sub-regions of the feature map for vectorization. . . . .	107
6.2	Diagram of the proposed SSM-VPR w/ Stage 1 rOSD Region Proposal, which takes a set of region proposals from the rOSD method developed by Vo et al. based on 2D pseudo-images from both stages and uses them to select sub-regions of the feature map for vectorization. . . . .	109
6.3	The SHM-VPR pipeline, here SSM-VPR stage 1 is kept the same as baseline but stage 2 now uses an estimated horizon line based on WaSR and projects it onto the feature map, the sliding window then moves along the map in a single row across the x-axis, using the y coordinate of the projected horizon line at each step. . . . .	110
6.4	Top Row: Initial PR Curve and AUC metrics for seven Plymouth Sound (Left) and Symphony Lake (Right) using Baseline SSM-VPR. Bottom Row: The same metrics after thresholding query images based on WaSR predicted land pixel percentage. . . . .	111
6.5	Overall PR Curves for my four pipeline versions on Plymouth Sound (Left) and Symphony Lake (Right). Baseline model and Unsupervised Region Proposal variants are consistent across both sets, SHM-VPR achieves greater performance on Plymouth Sound but lesser performance on Symphony Lake. . . . .	114



6.6	Example queries from the Plymouth Sound image set, divided into true positives, false positives and true negatives. Top Row: True Positive Queries. Middle Row: False Positives Queries. Bottom Row: True Negative Queries. . . . .	115
6.7	Example of Baseline SSM-VPR semantic regions: For each input image, the VGG16 activation map is divided into a set of sub-regions via sliding window. I show these feature maps in 2D by summing along the filter axis. . . . .	116
6.8	Example of SSM-VPR with semantic regions based on Selective Search: For this image, the activation is summed along the filter axis, then selective search region suggestions are made based on this to inform the extraction of feature map sub-regions. . . . .	117
6.9	Example of SSM-VPR with semantic regions based on the rOSD paper region proposal method: This figure is identical in terms of presentation to Figure 6.8. Interesting to note is that in the third column it can be seen that the rOSD algorithm always takes an even number of region proposals from both the Conv5_3 and Conv4_3 generated 2D pseudo images. . . . .	118
6.10	Figure inspired by the original paper [5]. A simplified representation of the spatial matching stage for a grid of query and retrieval vectors, taking a pair of anchor points between the two, their surrounding vectors along the row and column should also match if the features are spatially consistent . . . . .	119
6.11	A simplified representation of the spatial matching stage for a row of query and retrieval vectors extracted via the SHM-VPR method, taking a pair of anchor points between the two, their surrounding vectors along the row should also match if the features are spatially consistent. . . . .	120
7.1	The pipeline of Baseline SSM-VPR, as described in [5]. . . . .	124
7.2	SSM-VPR with SEFM applied to all feature maps passed through VGG16 for stages 1 and 2. This applies to both training images (reference) and query images. .	126
7.3	SSM-VPR with SALDR applied to the sliding window process of SSM-VPR stage 1. Feature map sub-regions designated by sliding window are projected onto a segmentation mask, if this area does not correspond to a minimum percentage of valid pixels it is discarded before moving on. . . . .	127

7.4	Our SHM-VPR alternative stage 2 method, using WaSR I extract a semantic edge belonging to the land class label, I then traverse the line to build a set of sliding window coordinates that can be projected back on to the stage 2 feature map. The sliding window goes across the x-axis, using the projected y-coordinate of each sampled point along the semantic edge. . . . .	128
7.5	PR-Curves with AUC values for each pipeline. Note that only valid queries are included in the calculation, many waterborne images contain no information (i.e. Open Sea) and are therefore not valid. Queries are determined to be valid if their land pixel percentage from WaSR segmentation map is 5% or more. . . . .	129
7.6	Recall@N percentage values for each pipeline. Only valid queries are included in the calculation. . . . .	131
7.7	Average Inference Time per query in seconds versus Recall@1 for each pipeline. Only valid queries are included in the calculation. . . . .	132
7.8	Top: A regular CNN feature map output for a given image. Bottom: A feature map produced using the SEFM method, for a given input a segmentation map is used to create a binary mask for the input, the original and masked inputs are both passed through the CNN and their feature maps are aggregated into a final output. . . . .	134
7.9	Top: Six input images examples. Middle: Regular feature maps generated by VGG16 Conv5_2. Bottom: Feature maps generated by VGG16 Conv5_2 using the SEFM segmentation method. . . . .	134
7.10	Top: An image produces a CNN feature map which is divided into a set of sub-regions via sliding window to be extracted and vectorized for nearest neighbour search as of a Deep VPR pipeline. Bottom: In addition to the feature map output, the image is also used to produce a segmentation map which is converted into a binary mask of valid/invalid class labels before semantic awareness is applied to filter valid regions. . . . .	135
7.11	First Column: Example image inputs. Second Column: Resulting CNN Feature map sub-regions selected for local vectorization via sliding window. Third Column: Same as second column but with sub-regions rejected by SALDR depicted as red squares. . . . .	136

7.12	The SHM-VPR alternative to SSM-VPR stage 2, instead of producing local vectors from a small sliding window across a VGG16 Conv4_2 feature map, we first take a binary mask based on a particular valid semantic class label and extract its upper edge. A row of windows are then localized along this edge and projected onto the feature map for local vector extraction. . . . .	136
7.13	Top: Six input images examples. Middle: CNN feature maps generated by VGG16 Conv4_2 for SSM-VPR stage 2. Bottom: Windows projected along a semantic land edge generated from each image to be used by SHM-VPR stage 2. . . . .	137
8.1	An example of how a user will see the Basecase Deep VPR image input and output during the survey . . . . .	142
8.2	An example of how a user will see the Deep VPR image input, output and Image Saliency heatmaps during the survey . . . . .	143
8.3	Left: An example interaction prompt from the human-in-the-loop survey, shows WaSR land segmentation in red. Right: If the user chooses ‘Reject’ on the left page, they are taken to a rudimentary image painting page where a new segmentation mask is created by highlighting parts in red. . . . .	144
8.4	From Jian et al. [10], “Trust scale items for human-machine trust and the corresponding cluster of trust related words on which they were based” . . . . .	147
8.5	Left: An example question from the Basecase Survey. Right: An example question from the XAI survey . . . . .	149
8.6	Left: Box Plots for Automation Bias ratio for all users across Scenarios 1-3 Right: Box Plots for Detrimental Algorithmic Aversion ratio for all users across Scenarios 1-3. . . . .	152
8.7	Left: Box Plots for Automation Bias ratio for all users across Human-Centered AI methods arms Right: Box Plots for Detrimental Algorithmic Aversion ratio for all users across Human-Centered AI methods arms. . . . .	153
8.8	Top Row: Box Plots for Positive User Reliance prompts (Fields 3-5) across Scenarios 1-3 Right: Box Plots for Negative User Reliance prompts (Fields 6-8) across Scenarios 1-3. . . . .	154
8.9	Top Row: Box Plots for Positive User Reliance prompts (Fields 3-5) across Retrieval Quality 1-3 Right: Box Plots for Negative User Reliance prompts (Fields 6-8) across Retrieval Quality 1-3. . . . .	155

8.10 Top Row: Box Plots for Positive User Reliance prompts (Fields 3-5) across  
Human-Centered AI methods arms: Box Plots for Negative User Reliance prompts  
(Fields 6-8) across Human-Centered AI methods arms. . . . . 157

# Chapter 1

## Introduction

### Contents

---

1.1	Introduction to Problem . . . . .	2
1.2	Motivations . . . . .	3
1.3	Objectives . . . . .	4
1.4	Overview . . . . .	5
1.5	Contributions . . . . .	6
1.6	Publications . . . . .	8
1.7	Outline . . . . .	9

---

## 1.1 Introduction to Problem

In the last decade, the use of autonomous systems for maritime navigation has seen a surge of interest, with well-established groups such as QinetiQ and the University of Southampton publishing the “Global Marine Technology Trends 2030” in 2015, which identified autonomous systems, big data analytics, and human-computer interaction as key technologies to be applied in the areas of commercial shipping, naval, and ocean spaces [39]. Since 2017, the Maritime Safety Committee (MSC), a subgroup of the International Maritime Organisation (IMO), has dedicated itself to guiding the international community on the research and development of autonomous systems for maritime navigation, with its first acknowledgment of autonomous systems on their agenda taking place in their 98th session [40], where the term “Maritime Autonomous Surface Ships” (MASS) was defined to refer to all vessels that make use of fully and semiautomated systems.

To aid this, various Computer Vision methods, a subset of Deep Learning focused on imagery, have been developed including collision avoidance [41] and fully automated navigation [42–44]. The former seeks to use technology to reduce maritime casualties and accidents, which, as identified by annual studies undertaken by the European Maritime Safety Agency (EMSA) [45], can result in consequences such as loss of human life, ship damage, and environmental pollution. Research into the latter seeks to build a connected system of autonomous subsystems that can mirror the navigational brain of a ship captain, something that Yan et al. refer to specifically as the “Navigation Brain System” (NBS).

Our key area of interest, which has not been tackled in previous research, is a Computer Vision task known as Deep Visual Place Recognition, which can use imagery of a vessel’s surroundings to retrieve matching examples from a known database, with these matched images providing previously logged positions to find out where a vessel is currently located without the need for systems such as GNSS, RADAR or LIDAR.

In this thesis we propose and develop a Waterborne Deep VPR pipeline intended for semi-autonomous decision support, providing increased vessel safety by providing approximate coordinates during GNSS downtime, such as when access to satellites is prevented or during more hostile encounters where some malicious actor has purposefully denied a vessel access to GNSS.

## 1.2 Motivations

A major motivation behind autonomous systems for maritime vessels is that of safety, for tasks such as collision avoidance the safety impacts are very clear, as part of the European Maritime Safety Agency (EMSA) annual studies between 2014-2022 most fatalities aboard maritime vessels were found to be a result of collisions, so by preventing these human life is preserved.

However, as seen in the world of autonomous cars, the idea of autonomous navigation in the mainstream tends to evoke skepticism [46], as people believe that AI could malfunction and cause accidents themselves. In the area of maritime navigation, Yan et al. [42] believe that by developing a potential Navigation Brain System through advanced technology to intelligently guide ships, they can actually become safer, a statement that is not unfounded as additional findings from the EMSA 2023 annual maritime incidents study found that 59.1% of accident events were the result of human error.

In addition to avoiding human navigational errors, in our introduction we briefly described situations in which Deep VPR could aid ships under a GNSS related attack to regain their correct location. GNSS attacks are unfortunately quite common, as GNSS signals are seen as easy targets [47], low signal strength makes them easy to block [48] and it is well known that civil GNSS signals lack encryption.

Two of the most well known GNSS attacks are Jamming and Spoofing [49]. GNSS Jamming involves an attacker sending out high-power radio frequency signals of similar frequency to the victims GNSS signal receiver, this prevents the victim from receiving authentic GNSS signals from orbital satellites, inhibiting their ability to navigate using said signals.

Spoofing on the other hand involves the sending of falsified GNSS signals to a victims receiver device, in order to trick them into believing they are located at a falsified position transmitted by the spoofer.

GNSS jamming can be countered by having backup systems during periods of attack, whereas spoofing could be countered by providing an additional reference point to compare against the vessel's current GNSS position to ensure consistency, this reference would then be periodically updated based on the visual surroundings of the vessel.

Finally, our objective is to gain insight into user reliance on Deep VPR systems in a maritime context, studies in the area of autonomous cars have shown that many people are still distrustful of AI when it comes to navigation [50] and this is still true for autonomous vessels, usually referred to as Unmanned Surface Vessels (USV) [51].

For USVs specifically, the main concerns are safety and reliability, particularly in terms of

collision avoidance as although USVs take out the risk of losing personnel on board, losing the vessel still incurs a significant monetary loss and as such low reliability turns away potential investors, which in turn slows adoption [52].

In this paper we are not proposing fully autonomous USVs, but rather onboard AI systems as decision support for non-GNSS based navigation, however, assuming that a user is putting their faith in our systems to navigate their ships, we are still beholden to ensure safe and reliable navigation, otherwise we risk the same slowing of adoption of such systems. As such, we will need to carry out a study to evaluate users willingness to use and rely on our systems in order to ensure that our research could see real world adoption.

As such, the targets of our user study will include both general AI users as well as a more specific target group of maritime navigators, as this is who the technology would be aimed at.

### 1.3 Objectives

#### **Compare existing Deep VPR methods against Waterborne Imagery to identify new image-domain based challenges**

Before heading straight into research and development of Waterborne specific Deep VPR, I first set out to compare known state-of-the-art Deep VPR methods across regular urban image domains versus the waterborne image domain. Outside of comparing quantitative performance in the form of Precision-Recall and Area Under Curve (AUC) statistics, I propose the use of Explainable AI in the form of image saliency to visually describe what these methods choose to be the most significant features within each domain such in order to exploit this later on.

#### **Develop a new in-house Waterborne Image set to facilitate the objectives of the paper**

At the beginning of this research, there were a low number of datasets fit for evaluating Deep VPR models in Shoreline localized waterborne areas, with more acceptable examples such as MaStr1325 and MODD2 [53, 54] unfortunately being more dedicated to obstacle detection specifically. To specify, a Deep VPR dataset should cover a specified area multiple times in the form of time separated runs across said area with a focus on local landmarks/-topology, as Deep VPR models must be able to locate the same places at different points in time, variance in terms of weather, lighting and obstructions also help to evaluate the models performance given such challenges.

As such, I worked alongside Marine AI Ltd., a USV development company situated in Plymouth, UK, to capture images from their Mayflower Autonomous Vessel built for IBM/Pro-mare of the Plymouth Sound area over seven runs to use as a new Deep VPR dataset.



**Develop a novel, waterborne specific Deep VPR model to achieve best performance on the image domain**

With key features and challenges of waterborne imagery identified through performance comparison and capture via the first two objectives, I develop and experiment with different novel Deep VPR model variants based off of existing state of the art by making modifications that exploit waterborne specific features. These include experiments with region proposal methods for designating image regions for conversion into useful local descriptors and using semantic segmentation of land features as knowledge prior for a suite of different beneficial modifications.

**Carry out a survey on user reliance on Waterborne Deep VPR**

I propose a two branched survey study with both a moderately sized group of anonymous AI informed users as well as a small group of known manned surface vessel navigators and operators, the first to measure general willingness to rely on our system when given the results bare versus examples where human-centered methodologies are applied, specifically with Explainable AI insight and Human-in-the-loop interactivity respectively.

The second branch with the small group of potential real world end users of this product will then go over these same sets of results produced by our system, with this moreso being used as a chance to carry out qualitative post-survey interviews about their thoughts on the system as a whole, whether or not they would consider adoption and if they believe either of our chosen human-centered approaches improved their experience and should be included in future.

## 1.4 Overview

Deep Visual Place Recognition utilising CNN based image representations has seen great success in recent years, surpassing classical hand-crafted techniques on a variety of benchmark datasets. However, these datasets all fall under a land-based domain, be it roads [55], railways [56] or indoor scenes [57]. However, as we have discussed Maritime Autonomous Surface Ship research has seen a surge in interest in the last decade, and as such it makes sense to test the viability of current Deep VPR state-of-the-art models for this domain as there are a multitude of inherent differences between the imagery taken on land and at sea.

As such, we seek to test the viability of Deep VPR on such imagery, first by applying state-of-the-art methodology to said images to get a grasp of initial performance; through this we find current open access image sets surrounding this domain to be somewhat lacking for

the purpose of Deep VPR training and evaluation. For example, at the beginning of the work making up our thesis, Symphony Lake [58] was one of the only waterborne datasets able to facilitate Place Recognition; however, it only covered a large lake area, which does not truly reflect the conditions one might expect of a larger area of shoreline.

To remedy this, in tandem with Marine AI Plymouth and the IBM/Promare Mayflower Autonomous Ship, we build a new in-house shoreline image dataset specifically for Deep VPR captured in Plymouth Sound UK. With this new dataset, we modify and evaluate existing state-of-the-art Deep VPR architecture with both region proposal techniques as well as other novel region-based representations in order to maximise performance on what is a much different type of imagery compared to land-based counterparts. By finding ways of adapting Deep VPR to this domain, we can take the first steps in creating a Deep Learning tool that can be deployed onto sea vessels to give an alternative form of navigation to GNSS when it is not available, as well as acting as a useful reference location to check current vessel GNSS against in order to detect spoofing. Both of these serve to aid ships by preventing them from navigating outside of safe operational areas and thereby avoiding maritime incidents.

Beyond the technical aspects, however, is the human element, if AI systems such as Deep VPR were to be proposed for sea vessels then trust needs to be fostered between the human user and the AI, we know from studies into autonomous cars that many people are opposed to the idea of handing navigational control over to AI systems so we believe it is important, even at this early stage of research into maritime autonomous navigation, to study how users interact with our proposed architecture and how different results and scenarios effect users willingness to rely on our AI system.

## 1.5 Contributions

The main contributions of this study can be seen as follows:

### **One of The First Evaluations of State-of-the-art Deep VPR on Waterborne Imagery**

During 2020-2022, excluding underwater imagery [59], we found that there was very little to no published research of Deep VPR being applied to waterborne imagery. As a result, our first publication, “Deep Visual Place Recognition for Waterborne Domains” [60], is likely one of the first published papers to present the applicability of SoTA to this domain. The work revealed new visual challenges compared to land-based benchmark datasets including water obstruction, variable distance from shore, lower atmospheric visibility and less stable camera

motion. When comparing quantitative performance, we found SoTA Deep VPR could retain high quality results, our novel explainable AI results also highlighted the types of features in waterborne imagery that received the most attention from the AI model as opposed to those of land imagery, giving us clues as to how to optimize performance in this domain.

### **Creating a New In-House Dataset for Waterborne Deep VPR**

In addition to the lack of widespread research on Waterborne Deep VPR, we also found that during this time there were seemingly no openly available datasets to facilitate training and evaluation of this task. Most Waterborne datasets were built for other tasks such as object detection and collision avoidance [53], also, those that could support Deep VPR research were often more bucolic in nature [58] rather than shoreline. To remedy this, we collaborated with Marine AI, a company in contact with our stakeholders UKHO, to create a dataset to facilitate Waterborne Deep VPR using multiple captured runs along the Plymouth Sound taken from the IB/Promare Mayflower Autonomous Ship.

### **Creating a Novel Horizon-Based Descriptor Extraction Method**

With our in-house dataset, we build upon the insights gathered from our first publication [60] by developing a novel implementation of the SoTA Deep Visual Place Recognition pipeline SSM-VPR [5]. Our first strategy was to focus on minimizing redundant feature extraction while maximizing salient feature extraction, we implemented this using two different region proposal methods for local descriptor extraction, but found that it was much more effective to instead have local descriptors be extracted along a relevant semantic edge. This latter method became the Semantic and Horizon-Based Matching for Visual Place Recognition (SHM-VPR) model, exploiting the tendency for salient land features to exist along natural land contours (i.e. The Horizon) and is presented in our second publication [61].

### **Further Application of Semantic Segmentation masks for Waterborne Deep VPR**

In order to gain a more significant performance boost from the Semantic Segmentation mask used to enable SHM-VPR and offset the segmentation models increase to overall inference time, we applied two additional segmentation techniques for enhancing Deep VPR and applied them to the SHM-VPR pipeline to create our Semantically Aware SSM-VPR pipeline. All methods take the same segmentation input and work independently of one another, providing different benefits in terms of precision, recall and inference time to create a

balanced overall improvement.

### **Analysing User Reliance on Waterborne Deep VPR**

Beyond technical contributions, we took inspiration from societal hesitations towards the adoption of AI based navigation, creating a study to measure user reliance through analysing themes of trust, confidence and technology dominance in our Deep VPR system as a navigational decision-support tool. The study revolves around the novel idea of measuring user attitudes toward Deep VPR while having access to different levels of human-centered techniques. This idea spawned three individual study arms including no interaction, explainable AI through image saliency and Human-in-The-Loop interaction. To promote a varied set of results, each arm presents examples of different ground truth levels along with different scenarios. To add additional validity and extract deeper insights, we also present our survey to five waterborne navigation domain experts and carry out a qualitative interview with each.

## **1.6 Publications**

The work in this thesis has contributed to several publications. The key contributions of each paper related to the main body of work are as follows:

### **L. Thomas, M. Edwards, A. Capsey, A. Rahat and M. Roach: Deep Visual Place Recognition for Waterborne Domains**

An Introduction of Deep VPR to Marine Images or “Waterborne Domain” with Performance and Explainable Visualization Analysis. Contributes to Chapter 5.

### **L. Thomas, M. Roach, A. Capsey, A. Rahat and M. Edwards: Semantic and Horizon-Based Feature Matching for Optimal Deep Visual Place Recognition in Waterborne Domains**

An attempt to optimize Deep VPR on a new in-house shoreline dataset using unsupervised region proposal and image segmentation. Contributes to Chapter 6.

### **L. Thomas, M. Roach, A. Capsey, A. Rahat and M. Edwards: Semantically Aware SSM-VPR for Waterborne Deep VPR**

A post-publication providing a large expansion on the incorporation of image segmentation into Waterborne Deep VPR to optimize performance using a knowledge prior to indicate

where “useful” features are located. Contributes to Chapter 7.

**L. Thomas, M. Roach, A. Capsey, A. Rahat and M. Edwards: Measuring User Reliance and Technology Dominance in Waterborne Deep VPR With Varying Levels of Interactivity**

A publication covering a survey and interview study tackling user reliance and the dominance of Deep VPR technology when given access to end users. Key themes include using AI under different scenarios, levels of quality and with additional levels of interactivity via Image Saliency XAI and novel Human-int-the-loop interaction. Contributes to Chapter 8.

## 1.7 Outline

**Chapter 2** Visual Place Recognition:

We introduce the background to current works in Visual Place Recognition, including modern and traditional methods for local and global based descriptor generation for VPR image retrieval.

**Chapter 3** Computer Vision Task Backgrounds and Applications for Waterborne Imagery

We cover several key Computer Vision Methods that have historically been used to enhance Deep Learning task performance including region proposal, saliency and segmentation, particularly within the domain of waterborne imagery.

**Chapter 4** Plymouth Sound Dataset:

We present a new in-house dataset containing images from several runs around Plymouth Sound to enable Waterborne Deep VPR training, testing and evaluation, built in collaboration with the IBM/Promare Mayflower Autonomous Ship.

**Chapter 5** Waterborne Deep VPR:

We present one of the first ever works on Waterborne Deep VPR by evaluating and comparing state-of-the-art performance on land and water imagery respectively. To highlight the difference in salient features, we present a novel image saliency technique for Deep VPR.

**Chapter 6** Improvement of Waterborne Deep VPR:

Building upon Chapter 5, we introduce the in-house shoreline dataset described in Chapter 4, allowing us to provide model evaluation on true shoreline imagery. To achieve greater performance, we apply both Region Proposal and Image Segmentation techniques to filter out sea and sky pixel information from local descriptor generation.

**Chapter 7** Semantic Segmentation based Knowledge priors for Waterborne Deep VPR:

We present an extended and more comprehensive semantically aware pipeline built upon our findings from Chapter 6, this chapter takes three independent yet complimentary image segmentation enhancement based techniques and applies them to state-of-the-art Deep VPR for maximum performance.

**Chapter 8** Measuring User Reliance and Technology Dominance in Waterborne Deep VPR With Varying Levels of Interactivity:

Having created an optimal Waterborne Deep VPR pipeline, we turn our attention to end users experience with said technology. In particular, we analyse user reliance and technology dominance when using Deep VPR as decision support, with standard, explainable and human-in-the-loop interactions.

**Chapter 9** Conclusions and Future Work:

We draw concluding remarks on the presented studies, looking to future use cases and the potential for development.

## Chapter 2

# Background of Visual Place Recognition

### Contents

---

2.1	Introduction . . . . .	<b>13</b>
2.2	Local Descriptor-Based Visual Place Recognition . . . . .	<b>14</b>
2.2.1	Local Descriptor Extraction . . . . .	14
2.2.2	Local Descriptor Matching . . . . .	17
2.3	Global Descriptor-Based Visual Place Recognition . . . . .	<b>19</b>
2.3.1	Global Descriptor Extraction . . . . .	19
2.3.2	Global Descriptor Matching . . . . .	20
2.4	Feature Descriptor Robustness to Perspective and Distance . . . . .	<b>22</b>
2.5	Dimensionality Reduction . . . . .	<b>24</b>
2.5.1	General-purpose Methods . . . . .	24
2.5.2	Image-based Methods for VPR . . . . .	26
2.6	Retrieval Refinement . . . . .	<b>27</b>
2.7	Evaluating Visual Place Recognition Models . . . . .	<b>28</b>
2.7.1	Definition of Ground Truth . . . . .	28
2.7.2	Metrics for Evaluation . . . . .	29
2.8	Methods for Real-time Localization . . . . .	<b>30</b>
2.8.1	Simultaneous Localization and Mapping . . . . .	31
2.8.2	Structure from Motion . . . . .	32

*2. Background of Visual Place Recognition*

---

2.8.3	Multi-Sensor Fusion . . . . .	33
2.9	User Reliance on AI for Autonomous Navigation . . . . .	<b>34</b>
2.9.1	Background . . . . .	34
2.9.2	Evaluating User Reliance . . . . .	35

---



## 2.1 Introduction

Visual Place Recognition (VPR) is a Computer Vision task that seeks to query images taken of a certain place in the real world and retrieve another image of the same place from a known database. For every image in this database, additional data associated with them is stored; specifically positional data such as a GNSS coordinate. By retrieving an image of the same place, one can assume they are located at the same GNSS coordinate, as such VPR has clear navigational applications in situations where GNSS navigation is not an option. Deep VPR is a more modern term that refers to pipelines combining this concept with Deep Learning methodology.

VPR can be seen as an extension of Image Retrieval, a computer vision task where various images are broken down into a set of descriptors, generally taking on the form of some mathematical feature vector(s), which are then matched to a neighbour image based on some distance metric (Often Euclidean) between their descriptors. Assuming these descriptors are robust such that similar values indicate the presence of shared visual features between images, those that have a minimal distance from the query image should contain similar content. The main use case for this is to find images that contain matching objects, also known as Content-Based Image Retrieval (CBIR).

VPR extends the use case to be used for image localization, whereas CBIR only aims to retrieve images containing the same content regardless of camera position, angle and distance when capturing the content, VPR seeks to retrieve images taken from the same location and as such similarities in camera position become more important.

Early applications of VPR originate from the field of robotics, where in order to localize an autonomous robot within a known reference map a capture of the surroundings can be taken, this is then matched to a previously taken image of those surroundings via image retrieval and, using a geo-tag associated with this retrieval, the robots location within the map can be approximated [62]. Research into this task has become highly relevant for both autonomous robots and cars, which can facilitate VPR through the addition of on board camera systems.

When performing VPR, descriptors are often representations of landmarks within an image, this can be seen as relating back to how we as humans navigate our surroundings by memorizing the locations of several landmarks to build up a mental map consisting of various landmark nodes [63].

Classic examples of VPR typically make use of hand-crafted methods in order to process images into feature vectors, however in the last decade Deep Learning models have now

surpassed these methods in terms of performance [64]. For VPR the Convolutional Neural Network (CNN) architecture is now the most widely used, as researchers have found that pre-trained CNNs built for image classification are fully capable of producing generalised image feature representations that can be used for numerous other tasks. Initial outputs from CNNs can be effectively transformed into feature vectors for the purpose of image retrieval through a series of state-of-the-art methods [5, 6].

In terms of challenges, locational images have an inherent set of highly variable conditions including lighting, time of day, weather and temporary obstructions. For example, one capture of a landmark may be taken when a truck passes by, whereas a second capture may not, these two images would, pixel-wise, become very different from one another despite being taken at the same location.

Rather than categorizing the different methods as handcrafted and Deep Learning, we instead categorize them by Local and Global descriptor-based methods as was done in Lowry et al.'s 2015 survey on VPR [62]. These categories give a better understanding of the ways in which data is used and compared across multiple VPR implementations, whereas the main difference between handcrafted and deep methods is simply how they extract data (algorithmic as opposed to convolutional).

## 2.2 Local Descriptor-Based Visual Place Recognition

As mentioned during our introduction, for VPR to be performed, input images must be converted into a vectorised feature descriptor format in order to perform retrieval via mathematically determining a set of nearest neighbours.

Classical state-of-the-art methods used for CBIR (and by extension VPR) would produce a set of local descriptors. Local Descriptors are essentially vectors that represent a subset of an image rather than the image as a whole, the most commonly cited example of an early local descriptor is the SIFT [65] keypoint descriptor.

### 2.2.1 Local Descriptor Extraction

The most common early example of local descriptor extraction is SIFT, the SIFT descriptor is a vector of fixed size that represents a set of features relating to a keypoint within an image, these features include surrounding scale, structure and rotational information. SIFT extracts these descriptors through a four stage process, starting by identifying scale-space extrema within

## 2. Background of Visual Place Recognition

---

several Gaussian convolved versions of an image to simulate varying scale space (including additional sets of images subsampled from the convolved image), then filtering out unwanted keypoints based on contrast and edge properties, using Taylor expansion for the former [66] and Hessian matrix for the latter.

Once the keypoint set is determined, for each keypoint the magnitude and direction of neighbouring pixels form an orientation histogram that is used to assign a dominant orientational value. Finally, the neighbourhood around the keypoint is once again taken into account and divided into a set of sub-blocks each of which is assigned an orientational histogram, the values of all these are then flattened and stored as the keypoint descriptor vector.

The methodology of SIFT based local descriptors presents information relating to notable objects in a way that is robust to variances in both scale and rotation; Other methods would build upon this in order to improve robustness to other variances such as lighting and colour.

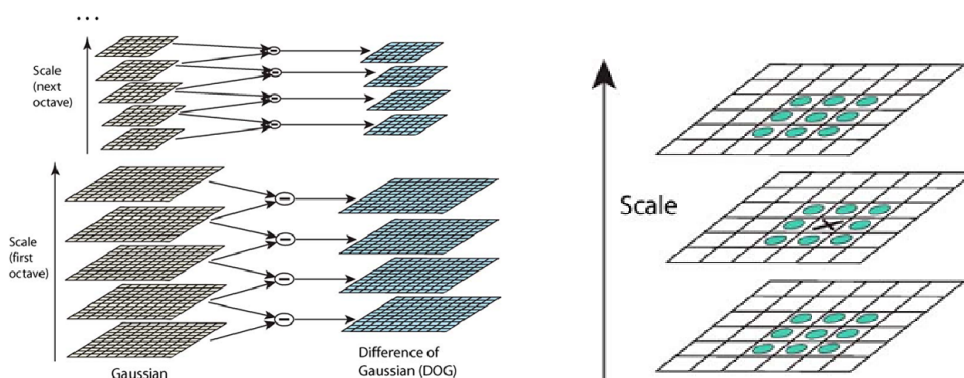


Figure 2.1: Left, the octave methodology used by SIFT to obtain numerous scale space images with neighbouring pairs subtracted to produce DoG. Right, SIFT finds local extrema via pixels with maximum and minimum values against both immediate neighbours and those from the above and below DoG's. Figures taken from [3].

After SIFT, the second most common example is SURF [4], whose development was mainly motivated by the comparatively slow computational speed of SIFT. SURF achieves a more efficient algorithm by using the integral image to identify regions of interest within which a sum of the Haar wavelet response can be used to build the feature descriptor.

SURF's region of interest detector employs a 'Fast-Hessian' detector that applies a box filter to the image that calculates approximations of the Gaussian second order derivatives, building a Hessian matrix which can be used to identify local maxima and points of interest.

By using the integral image, multiple box filters of increasing scale can be applied to the original image with the same computational time used for each rather than having to subsample

## 2. Background of Visual Place Recognition

---

the image.

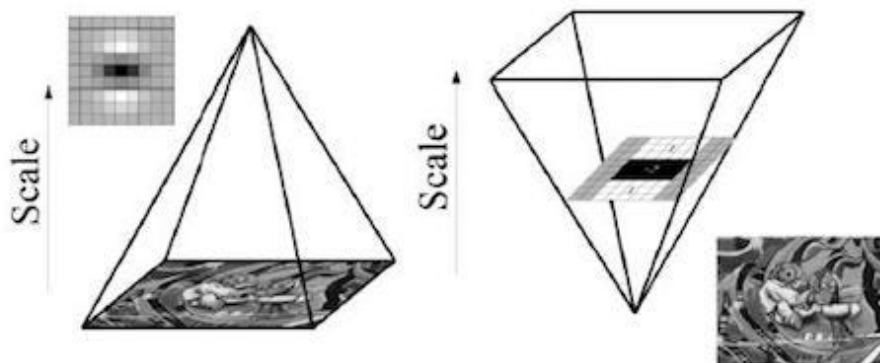


Figure 2.2: Depiction of how SURF box filter scaling differs from SIFT image scaling, on the left SIFT uses subsampling to analyse the image at different scales using DoG whereas on the right SURF changes the scale of it's Fast-Hessian detector box [4]

SURF descriptors are then generated for each detected keypoint using Haar-wavelet responses within the neighbourhood of the keypoint to assign orientation, after which descriptor components are built by selecting a square region based on orientation values surrounding the keypoint, subsetting this into 4x4 regions where Haar-wavelet responses for 5x5 features within the sub-regions are calculated and summed to form a 64 value feature descriptor.

For Deep VPR, most commonly cited state-of-the-art methods use global descriptors, however, many effective local descriptor methods have also been proposed. These often make use of region proposal techniques in order to determine sub-regions of an image to be fed to a CNN for the purposes of outputting a feature representation, such as Sunderhauf et al. [67] who pointed out that, compared to global descriptors, local descriptors can make a Deep VPR pipeline more robust to viewpoint changes and occlusion.

Sunderhauf's method specifically make use of Edge Boxes [68] to detect a set of possible landmarks within an image, each of which is passed through a CNN to generate a feature set. This feature set is dimensionally reduced using Gaussian random projection to create a set of local descriptors.

Other examples include Chen et al. [55] who make use of the later layers of VGG16 [69] (a popular pre-trained CNN) to identify salient landmarks by analyzing the most highly activated convolutional features within the output of the layer, which can be mapped back to a region of the input image. Descriptors can then be generated by pooling the convolutional features

## 2. Background of Visual Place Recognition

associated with the region into a single  $1 \times 1 \times C$  vector, where  $C$  is the number of channels of the conv layer.

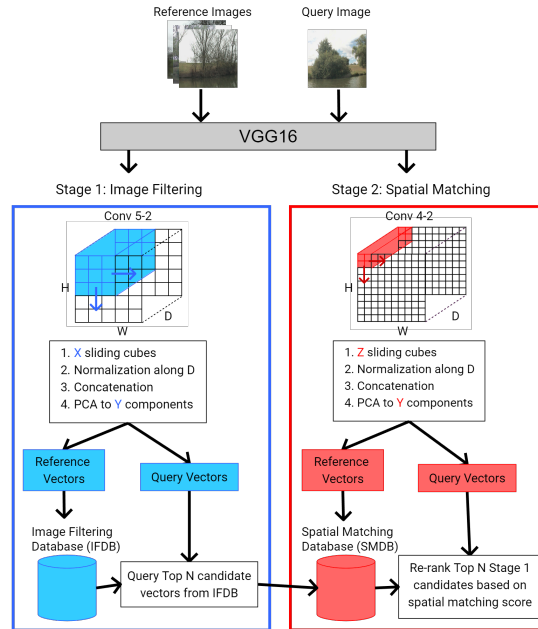


Figure 2.3: The SSM-VPR [5] pipeline which extracts sub-region based local vectors from CNN image feature maps in a two-stage approach.

More recently, Camara et al. [5] proposed a fixed region-based two-stage method SSM-VPR (see Figure 2.3), where a first set of fixed sub-regions are extracted from a later layer of VGG16 via sliding window and vectorized into local descriptors alongside an additional set of spatially aware local descriptors extracted from a higher layer using a smaller sliding window in order to build a more numerous set of fine spatial information based vectors. This second set is used as a refinement tool, being rearranged into the order in which they were extracted and compared against other secondary descriptor sets from an initial wave of nearest neighbours to check spatial consistency.

### 2.2.2 Local Descriptor Matching

Once descriptors are extracted from our database of images, they must be algorithmically matched to one another to identify neighbouring descriptors. For global descriptors, this step is simpler as each image would only be represented by a single vector and as such Euclidean distance can be used to find a nearest neighbour for a query image. However, with local de-

## 2. Background of Visual Place Recognition

---

scriptors, each image has a set descriptors that individually could be the closest neighbour to those of various other images, as such matching becomes more complex.

For classical methods (i.e. SIFT and SURF) Bag-of-features (BOF) [70] was employed, which involved assigning individual local descriptors to clusters representing visual words via K-means, these could then be turned into histograms that can then be used for distance-based searches such as Minkowski distance [71]. This method is largely inspired by text retrieval and uses a large vocabulary of visual words/clusters to produce each histogram, although one can also use a hierarchical vocabulary to improve search efficiency.

Chen et al. [55] proposed a framework for their previously discussed Deep VPR pipeline that followed this trend, computing clusters of visual words across all the local descriptors generated across the images of their database, with common words being downweighted compared to those that are less common, promoting features which are more rare and unique.

Sünderhauf et al. [67] and later Hou et al. [72] developed the QUT framework which was specifically designed to work with the Region Proposal enhanced Deep VPR models promoted by the former. QUT framework identifies bidirectional local descriptor nearest neighbours shared between a query image and other images from the database, for each pair a cosine distance is calculated along with a shape similarity measurement in order to penalise pairs whose width and height are not consistent with each other. The score for each database image can be seen as the summation of these scores for each local descriptor in the image that belongs to a bidirectional nearest-neighbour pairing with the query.

For SSM-VPR, Camara et al. [5] keep the local descriptors distinct values and instead sort all database images into a histogram, for each local descriptor from the query N nearest neighbours are found from the database via KD-Tree and their associated images receive a point on the histogram, this process repeats until all query descriptors have been covered and the histogram is sorted into descending order with the first N images being considered the overall nearest neighbours.

SSM-VPR then employs what is known as a retrieval refinement technique, whereby the spatial matching we discussed earlier is performed between the queries second set of local descriptors and those of each initial nearest-neighbour image. Each time a descriptor pair from these is found to be spatially consistent they receive a score on a new histogram for each of the N neighbours which is eventually used for re-ranking.

## 2.3 Global Descriptor-Based Visual Place Recognition

In contrast to local descriptors, global descriptors are more common in Deep VPR as opposed to classic handcrafted VPR methods, instead of representing subsets of an image through multiple descriptors the image is instead represented as a whole through a single global descriptor.

This is because the general Deep VPR pipeline consists of a pre-trained backbone CNN such as VGG16 and AlexNet trained on ImageNet, these backbones, initially built for the task of image classification, are able to produce robust image representations representing various generic objects and features from a wide variety of images and researchers have found that these representations can be made applicable to a number of other Computer Vision domains, including Place Recognition.

After passing through a backbone, CNN features are then typically aggregated into a single vectorized form to create a global feature descriptor which is later used for nearest neighbour matching.

### 2.3.1 Global Descriptor Extraction

One of the first examples of a global descriptor for Deep VPR was the Maximum activations of convolutions (MAC) vector [73], which for a CNN output of shape  $W \times H \times K$  where  $K$  is the number of filter maps, calculates the maximum activation within the region  $W \times H$  for each  $K$  and stores it as an element in the vector. This can essentially be viewed as a max-pool where the kernel is the same size as the output feature map.

Tolias et al. [74] extended this to work with the Regional MAC (R-MAC) [75] vector, which works in the same way but only represents maximum activations from a subregion of  $W \times H$ , Tolias et al. show that a simple summation of multiple R-MAC's can be an effective global descriptor.

In Babenko et al. [76], sum-pooled convolutional feature vectors, coined SPoC descriptors, were observed to also function as effective global descriptors, outperforming local SIFT key-point descriptors. The SPoC vector also uses PCA dimensionality reduction to reduce the size of the initial SPoC vector to improve matching performance.

Beyond the use of pooling-based aggregation for global descriptor extraction, one of the most commonly used state-of-the-art descriptors is NetVLAD [6]. NetVLAD is a generalised version of the existing VLAD layer that can be plugged into a CNN architecture and trained via backpropagation, when given a set of image features, NetVLAD learns  $K$  cluster centers and

## 2. Background of Visual Place Recognition

outputs a  $(D \times K)$ -dimensional vector that forms an aggregated representation of local feature descriptors, also known as a VLAD descriptor.

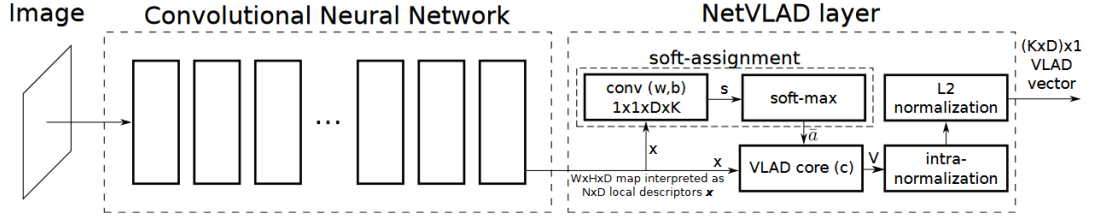


Figure 2.4: Figure depicting the NetVLAD pipeline, given an image a backbone CNN is used to output a feature map which is the used as input for the NetVLAD layer itself which soft-assigns the features to several clusters. Figure taken from [6].

The VLAD descriptor inherently has high dimensionality, making it expensive to compute, so in order to tackle this NetVLAD also makes use of a dimensionality reduction which is then followed by an L2 normalisation to get the final image descriptor.

NetVLAD is also an example of an end-to-end trainable pipeline for image retrieval/place recognition, which it achieves by using a modified version of the VLAD vector that, instead of hard assigning cluster centers via  $K$  nearest neighbours, instead uses a soft assignment where weights are applied to each input feature based on their distance to the closest cluster centre relative to all of the others, a convolution operation is then performed using these weights and softmax activation is applied to assign each feature to a cluster.

Using backpropagation, NetVLAD is able to tune these weights over time to ensure the generation of optimal global descriptors.

### 2.3.2 Global Descriptor Matching

Because global descriptors most commonly take the form of a single mathematical vector, matching can be achieved through the use of a distance-based metric and ranking a database of images in ascending order of such a metric.

The most common metric used by far is Euclidean distance [6, 7, 77], for two global descriptor vectors  $V_a$  and  $V_b$  of dimension  $N$ , Euclidean distance  $D(V_a, V_b)$  is simply the sum of the absolute difference between each pair of elements between  $V_a$  and  $V_b$  up to  $N$ :

$$D(V_a, V_b) = \sqrt{(V_{a_1} - V_{b_1})^2 + (V_{a_2} - V_{b_2})^2 + (V_{a_3} - V_{b_3})^2 + \dots + (V_{a_n} - V_{b_n})^2} \quad (2.1)$$



## 2. Background of Visual Place Recognition

More commonly represented by the compact Euclidean Norm term:

$$D(V_a, V_b) = \|V_a - V_b\| \quad (2.2)$$

Euclidean Distance therefore measures the overall dissimilarity between two global descriptors, as such when trying to identify a match we simply take the descriptor with the minimum distance value within the database from the query as this is assumed to be the most similar.

This can be seen in one of the early examples of deep VPR by Chen et al. [7], where global vectors are used to form a confusion matrix where each row represents a training image, each column represents a testing image and each element is the difference between CNN vectors for training and test images as shown in Figure 2.5.

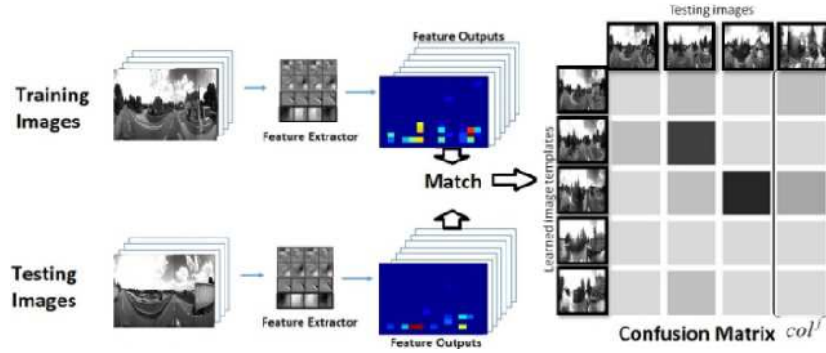


Figure 2.5: Generation of a confusion matrix for visual place recognition, features are extracted from each test image via CNN backbone and matched to those from all training images. Each matrix element  $M(i, j)$  represents the Euclidean distance between the  $i^{th}$  training image and the  $j^{th}$  testing image. Figure taken from [7].

The main drawback to this method is that the search is exhaustive, distance metrics between the queries global descriptor must be calculated for all database global descriptors, although this can be remedied through the use of more efficient search spaces such as the KD-Tree [78].

The reason Euclidean Distance is so dominant for global descriptor matching is likely due to two factors; It is a very simple algorithm, representing the minimum ‘distance’ between two objects of matching dimensionality within the space of that dimension and, because it can theoretically be applied to objects of any matching dimensionality, it is also extremely versatile for use with VPR methods that produce vectors of differing sizes.

However, Euclidean distance can become less effective as dimensions increase due to points becoming increasingly equidistant, an alternative method that works better for larger

dimensions is Cosine Similarity. As such, there are examples of Cosine similarity being used for global descriptor matching [74, 76].

Cosine Similarity works by taking two vectors of an inner product space and calculating the cosine of the angle between them, to determine if they point in a similar direction. The result of Cosine similarity is a value within the range  $[-1, 1]$  where 1 indicates that the angle of both vectors are proportional, 0 indicates they are orthogonal and  $-1$  indicates they are opposite.

The algorithm for calculating Cosine similarity simply involves dividing the dot product of two vectors,  $A$  and  $B$ , by the product of their lengths:

$$Sim(A, B) = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} \quad (2.3)$$

For VPR this can be applied to a database of images, retrievals can then be found by ranking results in descending order.

### 2.4 Feature Descriptor Robustness to Perspective and Distance

As was mentioned briefly during the coverage of local and global descriptor extraction, the reason these descriptors are extracted in the first place is due to their robustness to different changes in imagery.

Robustness is very important for VPR, as we need to not only match images containing the same content but to match them based on position, for which matching content is very helpful but we need to make sure we are viewing that content from a similar perspective and distance if we want to get the best positional match.

SIFT for example, as we covered earlier, uses keypoints that are invariant to scale, rotation and illumination, however it is only partially capable of handling perspective shifts. This is because during the process of clustering keypoint features, SIFT verifies each cluster using least-squares solution for an affine projection, which in the paper is described as effective for identifying objects within a 60 degree rotation relative to the camera [65].

SURF largely follows the same trend as SIFT, being aimed more at scale and rotation invariance, relying on the “overall robustness” [4] of the descriptor to somewhat cover skew and perspective changes which, in the paper, are considered to be second-order effects.

Therefore it is with Deep VPR techniques that we see an increase in the robustness of feature descriptors to perspective and distance, starting with the Sunderhauf method [67], where, as already discussed in Section 2.2.1, identifies a subset of landmarks in an image using

## 2. Background of Visual Place Recognition

---

Edgeboxes before passing them through the AlexNet architecture to layer conv3 to generate viewpoint-robust feature descriptors.

The claim that these descriptors are robust to viewpoint, is that by extracting invariant appearance-based convolutional features from proposed objects at different scales and locations, the method can extract discriminative information from the same landmarks at many variations of perspective and distance, which can be seen in Figure 2.6 from Sunderhauf et al.'s paper [67]. By extension, the Chen et al. [55] model also achieves this through a similar pipeline, replacing EdgeBoxes for a convolutional saliency-based region proposal algorithm.



Figure 2.6: The boxes in the figure above show landmark proposals extracted by EdgeBoxes that have been converted into convolutional feature descriptors for image matching. From both examples, you can see that the two image pairs were able to be matched from significantly different viewpoints.

Tolias et al.'s R-MAC based method also provides robustness to perspective and distance by producing several MAC vectors from the convolutional output of an image at different scales, then combining them into a single R-MAC vector via summation and l-normalization, the results of which can be seen in Figure 2.7. NetVLAD achieves a similar effect through the combination of local descriptors into a single VLAD descriptor.

The method which I have chosen to use as a base Deep VPR architecture for Chapters 5 through 8, SSM-VPR, achieves robustness through the use of a fixed set of region based local descriptors for both query and database images, as explained in Section 2.2.2, which are matched through a unique histogram system whereby each time a database image region is matched to a query, it receives a histogram score, with the largest scores being used to determine initial candidates.

This already improves robustness towards perspective, as content viewed from different



Figure 2.7: Multiple instances of the same building at different perspectives and distances matched to the query using the R-MAC representation.

perspectives can still be matched between imagery through this region-based score system, however it is the spatial matching stage of SSM-VPR that more so improves robustness to perspective, as it ensures the highest ranked candidate is that which is taken from the most spatially similar perspective to the query, making it more likely they are captured from the same position.

Camara et al. were able to show that this method outperformed many of the previously mentioned methods (Including Sunderhauf, NetVLAD etc.) on Berlin, Nordland and Gardens Point datasets which are known for having large perspective changes between image sets.

### 2.5 Dimensionality Reduction

Dimensionality reduction is a key component of many of state-of-the-art VPR pipelines. The task is to reduce high-dimensional data to a lower-dimensionality with minimal meaningful data loss, such that machine learning algorithms using said data retain similar performance [79]. In some cases, dimension reduction can even result in increased performance as a result of removing feature clutter.

This has obvious benefits with regards to computational time and data storage, lower dimensionality means less memory usage which allows Deep Learning models to assign larger batches of data to a machine's GPU for processing.

However beyond these surface-level benefits, these methods help to offset what is known as the "Curse of Dimensionality" [80], a phenomenon within Deep Learning where, given a fixed set of training samples, increasing the dimensionality of some output (i.e., a Fully-Connected layer) will initially increase performance until the model reaches a peak at which point further increase to dimensionality begins to reduce performance.

When increasing dimensionality, more samples are required in order to properly represent all possible samples that could occur within the feature space, otherwise the model will not be able to generalise itself to said space. If the model does not have enough samples to generalise then it will instead overfit onto the available samples, meaning test samples that differ from the overfitted data will likely receive poor results.

#### 2.5.1 General-purpose Methods

Outside of VPR specifically, there are a variety of methods for general purpose dimensionality reduction that can be applied to vectorized outputs within most fields of data science. The two

## 2. Background of Visual Place Recognition

---

most well known are PCA and Autoencoders.

Principal Component Analysis (PCA) [81] is a linear dimensionality reduction method that seeks to identify a set of principal components within the dimensions of a given input, then build a projection of the input based on these components, reducing dimensionality.

Principal components are identified through a series of steps, starting by a normalizing of all dimensions of the input to assign each one an equal weight, calculating a covariance matrix and using the eigenvectors and eigenvalues obtained from the matrix to determine the maximum variance of each dimension.

Assuming the user wants  $N$  principal components, the dimensions are sorted into descending order of maximum variance and the top  $N$  are assigned to be the principal components. Once completed, all input vectors can be truncated down to these  $N$  components.

PCA is a simple and effective method of dimensionality reduction, it is unsupervised meaning it can be fitted to any set of input data out of the box however it will only take into account linear relationships between variables.

For data with non-linear relations between it's dimensions, Autoencoders are a more effective method as they are based on the neural network model which is able to learn non-linear problems through training.

An autoencoder consists of three parts, the first is the encoder which is a series of neural network layers used to convert high dimensional data down to a dimensionally reduced vector at the second part known as the bottleneck, this is then passed to the final part known as the decoder which is a set of neural network layers that act as a reverse of the encoder, reproducing the original input from the bottleneck data.

This means that, by training an autoencoder to learn an optimal data representation at the bottleneck that minimizes the reconstruction loss, that being the difference between the encoder input and decoder output, the encoder can then be used post-training as a highly effective non-linear dimensionality reduction technique.

The drawback to using the autoencoder however is the need for this training process, in order to reach an optimal output you must have access to a large set of relevant training data otherwise you risk overfitting, which means the resources and time needed for using autoencoders are much greater than that of PCA.

### 2.5.2 Image-based Methods for VPR

In VPR, image descriptors need to have high enough dimensionality such that descriptors representing various locational imagery are able to cluster around those whose real world positions are close while also being clearly separate from those that are distant, even if the general features are similar (i.e. A Grassy Field, City Street, Motorway), by encoding specific landmark shapes unique to each location. However, having too high a dimensionality can make it difficult to form these locational clusters as more specific encoded features allow images to become further separated in the search space.

Many of the VPR pipelines discussed in Sections 2.2 and 2.3 employ some form of dimensionality reduction and those which do not often feature some aggregation technique that can be seen as loosely related, for example SIFT does not perform dimensionality reduction explicitly, however the SIFT keypoint descriptor does reduce dimensionality in some instances.

Consider that in SIFT a keypoint and its neighbouring features are initially represented as a subset of an input image of dimension  $H \times W \times C$  (Height  $H$ , Width  $W$  and Channel  $C$ ), SIFT converts this into a fixed size of  $4 \times 4 \times 8 = 128$ , depending on the size of the original subset, this could be seen as a reduction in dimensionality.

More importantly, each of the 128 elements within the descriptor represent relevant orientational values surrounding the keypoint, thus there is very little wasteful information being stored.

For Deep VPR, in Tolias et al. [74] MAC and R-MAC can both be seen as a form of dimensionality reduction as they are equivalent to performing a maxpool across the entire  $H \times W$  region (or a smaller  $h \times w$  subregion for R-MAC) of a feature map leaving only a vector of elements for each channel in  $C$ . By only focusing on the maximum response in each filter map along channel  $C$ , this method ignores noise within each filter map and each value is likely to represent the presence of a highly activated feature.

It can be argued that this is discarding a great deal of information from the CNN output; however, this is necessary as dealing with an entire  $H \times W \times C$  feature map would lead to a vector dimension in the thousands, if not tens of thousands.

The SPoC vectors proposed by Babenko et al. [76] are similar to MAC in that they use sum-pooling rather than max-pooling, however the Principal Component Analysis (PCA) [81] technique was also applied on top of this for further reduction and performance enhancement.

Despite these unique implementations, many state-of-the-art pipelines still make use of the previously mentioned PCA method to refine their descriptors, including NetVLAD [6] and

SSM-VPR [5].

For VPR, PCA projection can be seen as selecting variables across the set of descriptors that vary from image to image, identifying the presence of distinctive features as opposed to variables that show little variation and likely relate to some common feature found across most imagery that does not aid in distinguishing between examples.

### 2.6 Retrieval Refinement

So far we have discussed different processes of retrieving closest neighbours for both local and global descriptors and how these descriptors are aggregated/dimensionally reduced into optimal representations. For some VPR pipelines the final match is simply the top match according to these processes; alternatively, however, we can instead take a batch of the top  $N$  matches and perform what is known as retrieval refinement.

Masone et al. [82] go over a number of these methods in their 2021 Deep VPR survey, with the main four they identified being spatial verification, non-geometric re-ranking, query expansion, and diffusion.

Spatial verification can take a number of forms, but in general it consists of finding local feature-to-feature correspondences between query and candidate [83, 84] and using these to perform some form of affine transformation between the two images. From there, an algorithm such as RANSAC [85] is employed in order to find out how many inlier local features are present within the transformation between the images, the amount of which is summed and used as a re-ranking score.

This is useful for pipelines that employ global descriptor based pipelines, where spatial information is lost upon aggregating image features into a single vector. Spatial verification therefore allows us to analyse local feature-to-feature correspondences between individual image pairs for increased performance after a more efficient global descriptor comparison is used to filter down the entire retrieval set to a more manageable subset, keeping overall inference time shorter.

Non-geometric re-ranking appears to be an umbrella term defined by Masone et al. [82], referring to spatial verification without needing to use geometric correspondences. This list includes the reranking method employed by Toliás et al. [74] which uses approximate integral max-pooling to compare a MAC vector representing the query image to a set of subregions from the candidate image, returning the subregion that maximizes the similarity to the query, with this similarity being used as the re-ranking score.

Another example can be found in Yokoo et al. [86], where given a training image set that includes some form of class labelling, and a non-labelled image retrieval set referred to as an ‘index’ set, an offline stage can be performed before any online inference is done where each image in the index set is assigned a label from the train set based on soft voting from its  $k$  nearest neighbours. Then, at inference time, a query is a label through this same process,  $k$  nearest neighbours are gathered from the index set and those whose label id matches the query are moved up the shortlist of nearest neighbours.

SSM-VPR’s stage 2 [5] is another example of a non-geometric re-ranking method, where a set of vectors extracted from small sliding window subregions across mid-level convolutional features are stored for each image, such that, after a set of potential candidate images are retrieved from stage 1, these secondary sets of vectors can be compared between the query and each potential candidate in a process dubbed spatial matching.

SSM-VPR spatial matching arranges these vectors spatially to reflect the order in which they were originally extracted, anchor vectors are then identified between a query and candidate so that for each anchor pair, spatially equivalent vectors around these are compared in order to identify spatial consistencies. Every time spatially equivalent vectors around anchor points are found to be a closest match within both sets, the candidate receives a point, after all of these are checked across all anchor pairs the final score is used for re-ranking.

## 2.7 Evaluating Visual Place Recognition Models

### 2.7.1 Definition of Ground Truth

Before evaluating a Deep VPR model, true positive retrievals need to be properly defined. In content-based image retrieval, ground truth labels are based on the type of object seen in each image and as such retrieving an image with the same label is considered a true positive. Some global place recognition datasets do make use of ground truth labels, however these are usually associated with well-known landmarks (i.e. Eiffel Tower, Statue of Liberty etc.) contained within the image, acting as more traditional CBIR class labels [87].

When considering Deep VPR for navigational applications, the goal is to retrieve an image that is geographically within the same vicinity as the query. In Camara et al.’s [88] follow up paper on SSM-VPR, they define a metric known as “frame tolerance” that is, given a query image, it’s ground truths should be those image frames who were captured  $N$  instances before or after the query image frame, for the Gardens Point and Kudamm datasets where image



## 2. Background of Visual Place Recognition

---

captures along their paths were taken more sparsely (200 images total) the authors set a frame tolerance of  $\pm 2$  whereas for the much more dense Cities-8000 (8000 images) a tolerance of  $\pm 1$  was used.

This method is consistent as it means all queries have the same number of ground truths, it can also account for sudden changes in speed of the capture platform and therefore distance between a group of frames as although they would be further apart, they are still within the same frame tolerance threshold.

However, this method does not account for experiments using image sets with multiple overlapping runs, where the same locations are identified at different times, and therefore frames, meaning that assigning ground truth to images this way would become more complex.

The more common method is to define a ground truth region such that any image within  $D$  distance of a query is ground truth; however, the distance chosen must be informed by the density of images within the dataset capture area. For example, in the NetVLAD [6] paper, the authors evaluate their model against the Tokyo 24/7 dataset [89] which contains 76k training images across an area  $1600km^2$ , given that this is a relatively dense dataset and a distance of  $25m$  is assigned the ground truth.

In the Patch-NetVLAD paper [90] both methods are applied based on the nature of the dataset being evaluated, for Nordland the authors use a frame tolerance  $\pm 10$ , whereas for Tokyo 24/7 they use the same metric as in the original NetVLAD paper.

Generally speaking, camera orientation is not considered when determining ground truth within previous works [6, 88, 90], theoretically this could mean that locations with different features to the query which we would assume are wrong would actually be ground truth if they fall within the radius at a different orientation. However, considering that various benchmark evaluations ignore this, it seems safe to deem these instances as statistically insignificant.

### 2.7.2 Metrics for Evaluation

Once a method for determining ground truths is chosen, we can then proceed to evaluate our model in various ways, the most common method is measuring the true positive rate of the model, often referred to as accuracy or recall. This can be extended further by considering the true positive rate given  $N$  retrievals per query called *Recall@N*, where, if any of the  $N$  examples are ground truth, the query is considered to be recalled successfully and therefore true positive. When calculating the true positive rate, note that false negative in this case is any query that has not been recalled successfully, as such we simply divide the number of true

positives against the length of the query set.

The other most popular metric is mean average precision (mAP); however, this metric can only be applied to datasets where the number of ground truth images  $N$  per query is consistent:

$$\begin{aligned}
 mAP &= \frac{1}{Q} \sum_q AP_q \\
 AP &= \frac{R_q^{TP}}{R_q}
 \end{aligned}
 \tag{2.4}$$

Where  $Q$  is the total number of queries,  $AP_q$  is the average precision of the query  $q$ ,  $R_q^{TP}$  is the number of true positive retrievals for  $q$  and  $R_q$  is the total number of retrievals for  $q$ . When each query has ground-truth retrievals  $N$ , we can substitute  $R_q$  for  $N$  and refer to the equation as mAP@N.

Alternatively, if the number of ground truths per query is not consistent, we can instead use a metric developed in [91] whereby instead of calculating the AP per individual query, we instead calculate the AP across all retrievals simultaneously, thresholding the retrieval set according to some score or distance metric measuring their relevancy to the queries.

This is defined in [91] using both Precision and Recall metrics:

$$\begin{aligned}
 Precision &= \frac{\sum_q |R_q^{TP}|}{\sum_q |R_q|} \\
 Recall &= \sum_q |R_q^{TP}|
 \end{aligned}
 \tag{2.5}$$

By calculating these values at decreasing threshold intervals, we introduce more retrievals with lower relevance scores into the set until we reach the minimum threshold at which point we can plot the precision recall curve. From this, the ‘Area-under-curve (AUC) value can be used as a single value metric. Note that for traditional mAP or mAP@N, precision curves can be plotted by calculating mAP at different values of  $N$ , making sure that the top  $N$  per query are in order of relevance, thus allowing you to calculate AUC after normalising the recall values between 0 and 1.

## 2.8 Methods for Real-time Localization

Although not the focus of this paper, another highly relevant set of methods within the place recognition fields are methods for real-time localization - namely, Simultaneous Localization and Mapping (SLAM), Structure from Motion (SfM) and Multi-sensor fusion.

### 2.8.1 Simultaneous Localization and Mapping

SLAM is a method designed for fully autonomous navigation systems whereby a robot must navigate an unknown environment, for this to be achieved the SLAM method aims to both map the robots surrounding environment through on-board sensors while also localizing the position of the robot within this environment.

An important distinction to make between SLAM and VPR is that the former will not estimate a true, global position (i.e. GNSS) it is simply used to track a relative location based on the robots surroundings.

The method works by keeping track of the position of the robot,  $x$  (a localization and orientation vector), over time using the known movement vector of the robot,  $u$ , while identifying various local landmarks whose positions are assumed to be time invariant,  $m$  [8] (See Figure 2.8).

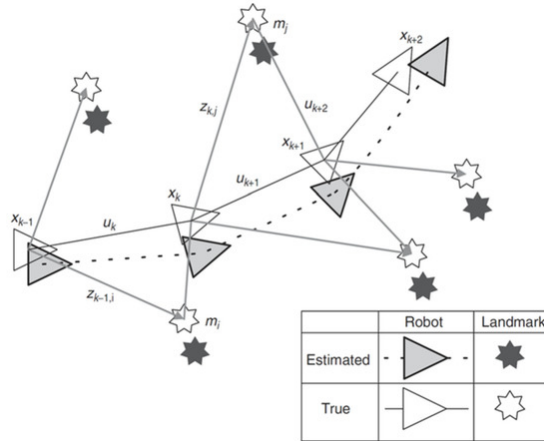


Figure 2.8: The SLAM problem as defined in [8], both the robot and landmark positions are mapped and localized simultaneously.

At each timestep, a vector containing observations from the robot of landmarks within  $m$  are recorded as an additional vector  $z$ , given all of this information SLAM attempts to predict the new position of the robot post-movement ( $u$ ) using the following probability distribution, where  $k$  is the current timestep and  $0$  is the origin:

$$P(x_k, m | Z_{0:k}, U_{0:k}, x_0) \tag{2.6}$$

The above equation can be modified based on the specific methodology and context of the problem, for instance some tasks may not use the motion vector  $u$  of the robot at all, in which

case the problem is entirely based on using the estimated positions of local landmarks.

### 2.8.2 Structure from Motion

SfM is defined as mapping the structure of a 3D environment using a collection of 2D images often combined with motion-based sensor information [92].

Most methodologies for SfM do this by generating a set of points within several images of the same scene, for example in [9] SIFT keypoints are generated for each image and the descriptors are used to determine potential pairwise correspondences (i.e. matching point pairs) between images.

This pipeline then uses Random sampling and consensus (RANSAC) to estimate the essential matrices, which define how points shared between two images relate, allowing for true corresponding pairs to be identified while also discarding outliers. Knowing how many correspondences there are between images, the pair with the highest amount will first be selected and a process known as bundle adjustment can be performed.

Bundle adjustment is a method used by many SfM algorithms that minimizes the re-projection error cost function, which consists of calculating the structure and camera calibration variables that produce the lowest discrepancy between the image measurements and their predictive model [93].

After calculating bundle adjustment for one pair, others can then be added in a greedy manner before recalculating the bundle adjustment until all relevant images are covered and we are left with a set of structured keypoints and camera variables, the result of which can be seen in Figure 2.9.

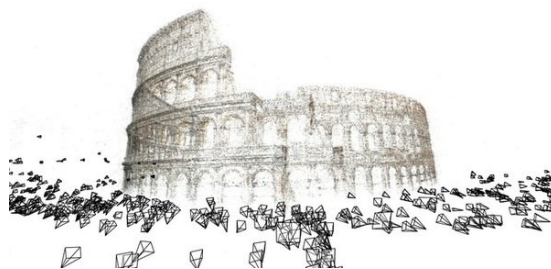


Figure 2.9: Results of structuring the coliseum in Rome using SfM [9].

In terms of how SfM relates to navigation, it can be applied as a purely image based method of performing 3D scene mapping as part of a SLAM pipeline, producing a large amount of land-

mark keypoints for predicting a robots location using timestamped footage from an onboard camera.

However historically, SfM is more commonly used within the fields of Geoscience [94] and Historical Preservation [95] as opposed to Computer Vision, where the point maps generated by SfM can be used to digitally recreate land surfaces and historical artifacts.

### 2.8.3 Multi-Sensor Fusion

As the title suggests, multi-sensor fusion is a term for any system that makes use of multiple types of sensor information, such as Camera feeds, LIDAR, RADAR, etc. in order to achieve some singular goal (i.e. autonomous navigation) [96]. Using multiple sensors allows such systems to make up for the limited perceptual capability of each individual sensor [97].

In Yeong et al.'s 2021 survey of the field, three approaches are identified; High-level Fusion (HLF), Low-level Fusion (LLF) and Mid-level Fusion (MLF). HLF allows for each individual sensor to carry out tasks, such as object detection and tracking, independently of one another before fusing this information later in the pipeline, most often using a Kalman filter [98].

The Kalman filter is a widely used technique for navigational systems that uses measurements over time to predict the current state of a system. In the case of Multi-sensor fusion, these measurements would consist of object detections from camera/RADAR/LIDAR which the filter uses to produce a state that minimizes the level of uncertainty of these measurements.

HLF with Kalman filter is a lower-complexity approach but suffers when one of the sensors is unavailable, inhibited or is reporting low confidence (i.e. Camera stops seeing ahead in bad weather) as there are now less measurements to work with for accurate state prediction.

MLF is similar to HLF only that instead of providing measurements from each sensor individually, their intermediate outputs (i.e. RGB image from a camera and locational information from RADAR/LIDAR) are fused before being passed through some classification method [99].

LLF goes a step further, taking the immediate raw outputs from each onboard sensor and fusing them into some sort of computational unit, this approach is the most complex implementation-wise and requires precise calibration of all sensors involved as to not introduce offsets between them.

Multi-sensor Fusion is a popular method for production level autonomous cars such as Tesla's AutoPilot [100] and Waymo (Formerly known as the Google Self-Driving Car Project) [101], the latter of which is employed as a taxi service in San Francisco, Phoenix, and Los Angeles. However, they also require heavy investment due to the large upfront costs of high-

resolution camera systems, RADAR and LIDAR as well as maintenance of these over time and as such are not an immediately available alternative to purely Computer Vision-based navigation techniques.

## **2.9 User Reliance on AI for Autonomous Navigation**

### **2.9.1 Background**

As this work proposes the creation of a Deep VPR model for decision-support in maritime navigation, the end result can be seen as supplementary to a fully our semi-autonomous system, for example, in real-time methods such as SLAM, a common error one may encounter is that of error drift [102], where small localization errors overtime result in an offset later on during traversal. Deep VPR is often used as a tool for loop closure, which is when we encounter a previously seen location and a discrepancy is detected between current position and previous position at this location, the system can attempt to correct the error [103].

However, autonomous navigation is a hot topic among end users, with many showing distrust towards such systems [50], which can dissuade adoption as users are unwilling to rely on them. Because this tool may be used as an autonomous aid in maritime navigation, it is important that we cover such studies into user trust in autonomous navigation so that we can design our model in a way that maximizes positive outcomes based on the level at which the user relies on such a system.

Reliance, which relates to feelings of trust, confidence and dependence, is a term that broadly speaking is used to describe the extent to which someone requires or feels the need to use some resource to function. In the context of AI, when someone relies on a system, they are often giving away personal agency with the expectation that the AI can perform some task to a similar or greater level of effectiveness [104].

However, we must also be aware of the dangers of making users rely on such technology - as no AI can be 100% accurate, it will ultimately make some mistakes and in these cases if a user continues to rely on the system then errors will occur. This concept is referred to as technology dominance [105], meanwhile the opposite would be referred to as technology avoidance and can also result in sub-optimal results as humans themselves can make mistakes that the AI would not, therefore when studying reliance we believe it is important to strike a balance that maximizes the number of true positive outcomes of some task.

The most common studies related to user reliance on autonomous systems focus on the

## 2. Background of Visual Place Recognition

---

emerging field of autonomous cars, with companies such as Waymo offering autonomous taxi services in places such as California [106]. In such studies, one of the factors that contribute to technology avoidance or distrust include an unwillingness to give up control [50], which some researchers agree is a justifiable reason not to fully rely on AI as removing human agency can cause human users to lose skill and develop unwanted habits overtime that may cause accidents when placed back in a manual vehicle [107].

Another factor is legal issues in case of accidents [108], however in the United States new car legislation in 2016 deemed autonomous cars such as the Google AI to be drivers, making the case here more concrete [109]. The EU also has published new legislation regarding autonomous cars, with only level 3 autonomous vehicles, that being those with a designated safety driver, being allowed on public roads [110].

As such, studies are still most commonly aimed at studying user interaction rather than things such as monetary benefits/drawbacks. To our knowledge, there has not yet been another paper that approaches this field from the angle of Deep VPR as decision-support.

### 2.9.2 Evaluating User Reliance

As the definition of reliance often overlaps with the more complicated topic of trust, we believe that methods designed to evaluate the latter in the context of AI can also give insight into the former. This is in line with previous research, who suggest that trust is has a direct correlation with users willingness to rely [111].

One of the most well cited works on analysing user trust in autonomy is Jian et al.'s paper "Foundations for an Empirically Determined Scale of Trust in Automated Systems" [10], in this work the authors set out to produce a method of directly measuring user trust in autonomous systems, the results of which allow for a robust assessment.

This was carried out by providing 126 university students as participants to rate a series of words on how much they believed them to relate to trust or distrust in general, between humans and between human and machine. The authors then calculated the average trust rating and distrust rating for each word, from which a strong negative correlation between the two opposites was found, validating the assumption that they are indeed opposites.

With words scaled from distrustfulness to trustworthiness, the researchers organized these into bins based on their average rating from which phrases relating to the words were developed in order to be used as prompts for a likert scale, where users can mark down how much they agree with the phrase (See Phrases in Figure 2.10). These scales are now some of the most

## 2. Background of Visual Place Recognition

---

well-cited and used for measuring human trust in AI.

Item	Words Groups from Cluster Analysis
The system is deceptive	Deception Lie Falsity Betray Misleading Phony Cheat
The system behaves in an underhanded manner	Sneaky Steal
I am suspicious of the system's intent, action, or output	Mistrust Suspicion Distrust
I am wary of the system	Beware
The system's action will have a harmful or injurious outcome	Cruel Harm
I am confident in the system	Assurance Confidence
The system provides security	Security
The system has integrity	Honor Integrity
The system is dependable	Fidelity Loyalty
The system is reliable	Honesty Promise Reliability Trustworthy Friendship Love
I can trust the system	Entrust
I am familiar with the system	Familiarity

Figure 2.10: Results of grouping words ranging from high average distrust association to high average trust association into related phrases [10]

Going back to reliance specifically, a new method of evaluation the method for measuring Technology Dominance proposed by Cabitza et al. [105]. The main goal of this measurement is to find a balance between human-AI interaction such that as many positive outcomes as possible can be achieved, the authors hypothesized that in any given task assuming both human and AI can make mistakes and that instances in which these occur for each party do not necessarily overlap, too much dominance in either actors favor will result in a sub-optimal performance compared to a mixed approach where the two correct eachother whenever possible.

To measure this, two scales were suggested, Automation Bias and Detrimental Algorithmic



## 2. Background of Visual Place Recognition

---

Aversion. These terms describe the two sides of the previously mentioned hypothesis, with both acting as odds ratios, with Automation Bias being the ratio over  $N$  cases of likelihood of detrimental over-reliance (dor), number of cases where reliance on AI results in a negative outcome, versus beneficial self-reliance (bsr), the number of cases where self-reliance results in a positive outcome, for this odds ratio values over one indicate the AI may be inducing a negative outcome on the users decision making.

$$AutomationBias = \frac{dor}{N - dor} \frac{N - bsr}{bsr} \quad (2.7)$$

Detrimental Algorithmic Aversion follows a similar formula, being the odds ratio between detrimental self-reliance (dsr), the number of cases across  $N$  where self-reliance lead to a negative outcome and beneficial over-reliance (bor), cases where reliance on AI lead to a positive outcome.

$$DetrimentalAlgorithmicBias = \frac{dsr}{N - dsr} \frac{N - bor}{bor} \quad (2.8)$$

This work is highly relevant to ours as it was developed specifically within the decision-support framework, as such it is an effective modern metric for studying reliance on Deep VPR.

## Chapter 3

# Computer Vision Task Backgrounds and Applications for Waterborne Imagery

### Contents

---

3.1	Introduction . . . . .	<b>40</b>
3.2	Object Detection/Region Proposal . . . . .	<b>41</b>
3.2.1	Handcrafted Methods . . . . .	42
3.2.2	Deep Learning Methods . . . . .	45
3.2.3	Applications in Waterborne Imagery . . . . .	48
3.3	Image Saliency . . . . .	<b>50</b>
3.3.1	Handcrafted Methods . . . . .	51
3.3.2	Deep Learning Methods . . . . .	53
3.3.3	Applications in Waterborne Imagery . . . . .	59
3.4	Semantic Segmentation . . . . .	<b>62</b>
3.4.1	Fully Convolutional Networks . . . . .	63
3.4.2	Deconvolutional Networks . . . . .	65
3.4.3	ResNet-based Fully Convolutional Networks . . . . .	68
3.4.4	Applications in Waterborne Imagery . . . . .	71
3.5	Human-Centered AI . . . . .	<b>72</b>
3.5.1	Explainable AI . . . . .	73

*3. Computer Vision Task Backgrounds and Applications for Waterborne Imagery*

---

3.5.2 Human-in-the-Loop . . . . . 75

---

### 3.1 Introduction

Computer Vision is an interdisciplinary field that enables computers to extract information and data from real world imagery. This can include 2D or 3D Images, 3D Scans/Pointclouds and Video Sequences. The motivation is to allow machines to view the world similarly to a human identifying various objects within its field of view and making decisions based on their properties.

This can be achieved using hand-crafted or Deep Learning frameworks, with the latter being more recent. In particular, the Convolutional Neural Network (CNN) architecture is a popular choice for Deep Computer Vision, allowing for efficient extraction of generalizable high-level activation maps, which can be condensed into a vectorized output using Dense Neural Network layers. Examples of Computer Vision tasks include:

**Image Classification.** A task that seeks to classify an object or a set of objects within an image via numerical outputs representing learned “class” labels, where each class is a type of object (i.e. 1 = 'Dog', 2 = 'Cat', etc.). Models are often trained to output a predicted bounding box that is assumed to contain the labeled object.

**Object Detection/Region Proposal** This task seeks to detect a number of notable objects within an image and output a bounding box for each. The goal of this task is not to classify objects with a label, but to quickly and efficiently detect the location/presence of generic objects of interest. Use cases include collision avoidance and as a knowledge prior which one may leverage by enabling some other model (i.e. Image Classifier) to focus only on the detected regions.

**Content Based Image Retrieval.** Covered briefly in Chapter 2, CBIR involves extracting data from imagery and aggregating the data into an effective numerical vector representation, often called a descriptor. Images containing the same content should have numerically similar descriptors, such that other images containing the same content can be retrieved via a distance-based descriptor search.

**Image Saliency.** Attempting to mimic the behaviour of the human visual system in order to focus on the most important information within the scene, image saliency tasks aim to identify segmentations that can separate background features from important objects within a scene. In an Explainable AI (XAI) context, image saliency can instead be seen

as a description of what objects are seen as being important to the AI model during its decision making, helping to explain this process to the user.

**Semantic Segmentation.** Beyond simple Image Classification, this task seeks to associate a class label with every individual pixel in an Image, tasks vary from detecting and segmenting objects such as cars and humans from a street view to classifying portions of waterborne imagery into categories of land, air and sea.

Many Computer Vision models are designed to solve single tasks such as those described above, however in many instances these are instead simply a sub-task of some larger task. For example, an object detector for cars may be used over a parking lot to measure the number of spots currently occupied whereas a semantic segmentation output could be used to inform a separate image classification CNN.

One of the many areas in which Computer Vision can be helpful in is tasks related to Waterborne Imagery, including Maritime Surveillance [17, 27, 112, 113] and Collision Avoidance [37, 114], where equivalent land-based approaches are inadequate due to unique artifacts found in Waterborne Imagery including water obstructions, reflections and wakes.

In this Chapter, we cover a set of Computer Vision tasks with applications for enhancing Waterborne Imagery, beginning with a brief background of the task as a whole and it's existing implementations, then discussing previous examples of their use for waterborne-based image tasks.

We also briefly cover the use of Human-Centered AI in the form of Explainable AI and Human-in-the-loop for Computer Vision, which are topics related to the end user experience relating to such systems after they are deployed, particularly regarding the black-box nature of CV outside of AI research.

## 3.2 Object Detection/Region Proposal

As we mentioned briefly in the Introduction section, Object Detection is a task whereby we attempt to identify and localize a set of notable objects within an image via labelling and bounding box outputs. Motivations for the implementation of such a method includes Face Recognition, which has applications within both commercial and law enforcement domains [115, 116], Pedestrian Tracking to inform the interactions and movements of autonomous systems [117] and Vehicle Tracking for use in emerging smart cities and smart traffic systems [118].

### 3. Computer Vision Task Backgrounds and Applications for Waterborne Imagery

---

The inherent challenge of performing these tasks is being able to identify objects under various conditions and circumstances. Lighting conditions and camera angle can have a big impact on the clarity of human faces within an image for example, meanwhile pedestrian detection must be able to identify humans in various poses while also dealing with visual clutter as seen in Figure 3.1.

Region Proposal is functionally similar to Object Detection and whichever definition is applied to a task or model typically depends on the context. Compared to Object Detection, Region Proposal is more concerned with identifying a set of bounding boxes that are *likely* to contain an object or notable features in general.

Generally speaking, methods designed for Object Detection can also be used for Region Proposal and vice-versa, however many Deep Object Detectors are trained in a supervised manner to detect a specific set of object classes and as such may not translate well to generic Region Proposal.



Figure 3.1: Examples of pedestrian detection training images showing the large variety of poses, illumination and clutter that one could encounter when trying to perform detection. Figure taken from [11].

#### 3.2.1 Handcrafted Methods

Some of the first major implementations of Object Detection include the Viola-Jones object detection framework [12] and the histogram of oriented gradients (HOG) [11], both of which were handcrafted feature representations made before the popularity of deep learning-based methods [119].

The first of these, Viola-Jones, was made for face detection using a large set of rectangular feature detectors reminiscent of Haar Basis functions which were divided into two to four sections. For each feature, the sum of pixels within binary labelled subsets of these sections be subtracted eachother to perform edge detection, from the initial set, a smaller set of highly effective rectangular features were selected with AdaBoost [120], which was also used for training.

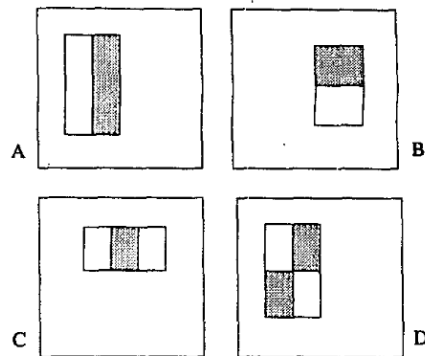


Figure 3.2: Examples of rectangular features used by Viola-Jones method [12]

Viola-Jones used a detection cascading approach, where multiple classifiers of increasing rectangular feature numbers would be trained and used to produce true positives and false positives which could then be sampled from as true positives and true negatives for training the next classifier.

Histogram of Oriented Gradients (HOG) is a detector designed for pedestrian imagery. HOG makes use of gradient magnitude and angle maps calculated over an image which are then divided into a set of  $8 \times 8$  pixel groups referred to as “blocks”, these blocks are used to contribute to a 9-bin orientational histogram similar to SIFT [65], each bin in the histogram represents a  $20^\circ$  angle, with pixel values being accumulated based on their magnitude and angle map values.

HOG introduced an additional step whereby overlapping  $2 \times 2$  groups of these 9-bin histogram calculated from the previous step are contrast normalized and collected to form the HOG descriptors, making the implementation more robust to illumination and background-foreground contrast change.

For methods designed more so for Region Proposal, Selective Search is a popular method that combines typical exhaustive search with segmentation. In order to capture objects at all scales, the method begins by applying the Felzenszwalb and Huttenlocher “Efficient graph-based image segmentation” [121] method to the image to produce the initial set of regions. After, a greedy algorithm is applied to the region set, calculating the similarity between all neighbouring regions and merging the two most similar regions together and adding them to the region set, then recalculating the similarity between this new region and its neighbours before repeating the process indefinitely until the image is reduced to a single region (Figure 3.3).

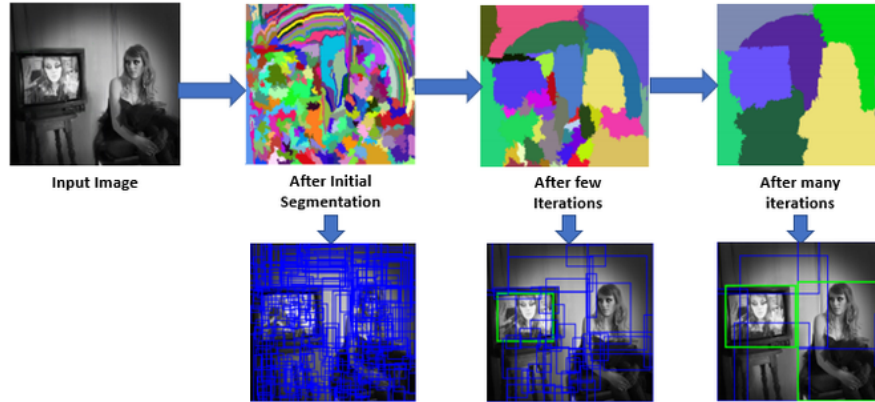


Figure 3.3: Visual depiction of the Selective Search exhaustive segmentation followed by several merges over the iterations, below each segmentation image are the resulting bounding-boxes [13].

The similarity metric used between regions consists of four main parts, each of which needs to be propagated hierarchically as regions are merged by the algorithm such that the image pixel values do not need to be accessed. These four metrics that make up the overall similarity  $s(r_i, r_j)$  between a region  $r_i$  and  $r_j$  are  $s_{colour}(r_i, r_j)$ , which record initial one-dimensional colour histograms for each region,  $s_{texture}(r_i, r_j)$ , representing texture similarity via SIFT-like binned orientational histograms for each colour channel,  $s_{size}(r_i, r_j)$ , the fraction of an image that  $r_i$  and  $r_j$  jointly occupy, finally, the fourth component,  $s_{fill}(r_i, r_j)$ , measures how well  $r_i$  and  $r_j$  fit into each other.

The final similarity metric  $s(r_i, r_j)$  is as follows, where  $a_i$  is either 0 or 1 based on whether or not that particular component is being considered:

$$s(r_i, r_j) = a_1 s_{colour}(r_i, r_j) + a_2 s_{texture}(r_i, r_j) + a_3 s_{size}(r_i, r_j) + a_4 s_{fill}(r_i, r_j) \quad (3.1)$$

we also have EdgeBoxes [68] and randomized Prim [122]. EdgeBoxes evaluates bounding boxes over a Structured Edges [123] output based on an input image, the resulting edge map is divided into a set of edge groups, each of which is evaluated against a set of bounding boxes. For each box, the portion of the edge group that falls exclusively within the box is taken into account during evaluation while also measuring the portion of edge pixels belonging to groups that overlap the current box as a negative to the overall score. The intuition for EdgeBoxes is that the number of contours that are wholly contained in a box increases the likelihood that the box contains an object.

Randomized Prim on the other hand makes use of partial spanning trees built over a graph based on superpixel segmentation of an input image. Initialising on a single superpixel, the



algorithm iteratively builds up a partial tree by adding neighbour superpixels nodes based on the value of their edge weight, which is calculated through a combination of measuring color similarity, common border ratio and size.

### 3.2.2 Deep Learning Methods

Later Object Detection methods would leverage the Convolutional Neural Network (CNN) model, the most well-known method being the Regional Convolutional Neural Network (R-CNN) family, including the original [14] and its two major follow-ups Fast R-CNN [15] and Faster R-CNN [16] which improved the models efficiency for the Object Detection task.

R-CNN (See Figure 3.4) is built using two major parts, Selective Search [13] for unsupervised Region Proposal and a CNN pre-trained on a large auxiliary Image Classification dataset such as ImageNet [124]. Each region proposal takes the form of a bounding box surrounding what Selective Search believes to be some generic object, this bounding box region is then rescaled into an individual image to be passed through the CNN to be classified.

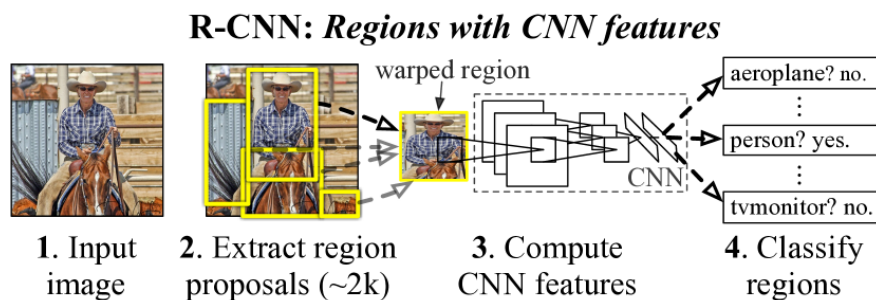


Figure 3.4: The R-CNN pipeline [14], which uses Selective Search for region proposals that are then warped and passed through a CNN for individual object classification.

The biggest issue with R-CNN when it came to computational efficiency was that each proposed region had to be fed to the CNN as an individual image, Fast R-CNN (Figure 3.5) addresses this by instead passing the entire input image through the CNN, then projecting the bounding boxes of the proposed regions onto the intermediate convolutional feature map. These sub-sections of the feature map are then ROI pooled into consistent sizes and passed to the final Dense layers for classification.

Faster R-CNN does away with using a pre-built Region Proposal method (i.e. Selective Search) and instead uses a Region Proposal Network (RPN). The RPN shares convolutional layers with the CNN such that once the final shared layer is reached, before the R-CNN performs any classification the RPN will apply sliding windows of differing shapes over the con-

### 3. Computer Vision Task Backgrounds and Applications for Waterborne Imagery

---

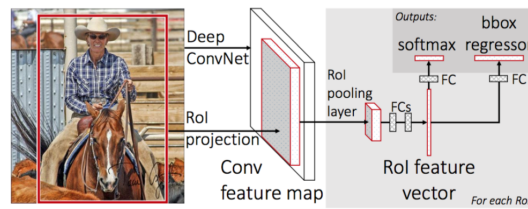


Figure 3.5: Fast-RCNN pipeline [15], which does away with the image warping stage and instead projects RoI's onto the CNN feature map, shape consistency is then ensured via the RoI pooling layer.

volutional feature map that are used to regress both a probability value for the presence of an object as well as storing the coordinates of the window as a bounding box output.

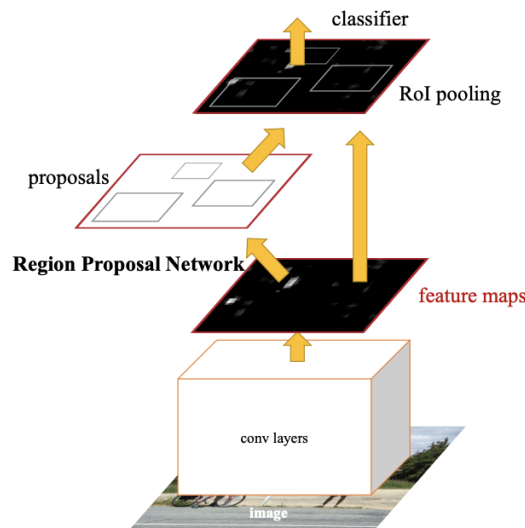


Figure 3.6: Faster-RCNN pipeline [16], using a trained RPN to generate proposals based on feature maps which are RoI pooled and classified.

This RPN does not work out-of-the-box, but this is solved by training the Dense output layers it uses end-to-end through backpropagation and stochastic gradient descent (SGD) [125]. For the task of general Object Recognition this training was facilitated by the availability of the PASCAL VOC 2007 [126] dataset, with more recent pre-trained versions using more up-to-date alternatives such as ImageNet [124].

For other computer vision applications, specifically visual place recognition, where object classes of interest may be different from those of image classification, these supervised methods can be difficult to incorporate, whereas an unsupervised method can be applied to any

### 3. Computer Vision Task Backgrounds and Applications for Waterborne Imagery

---

domain.

For this paper, designing and labeling a new set of bounding box classes for our waterborne image data would likely constitute an additional body of work on top of the contents covered in Chapters 5-8, as such it was not deemed feasible within the allotted time frame to train a Region Proposal Net.

In addition, because popular RPN's such as Faster R-CNN are pre-trained on datasets such as ImageNet, which focuses on smaller individual objects we believed they would not translate well to broad shoreline-based features and as such we opted to make use of unsupervised methods.

A more recent Deep Learning based method that can work without training can be found in Vo et al.'s work, "Toward Unsupervised, Multi-Object Discovery in Large-Scale Image Collections" [127].

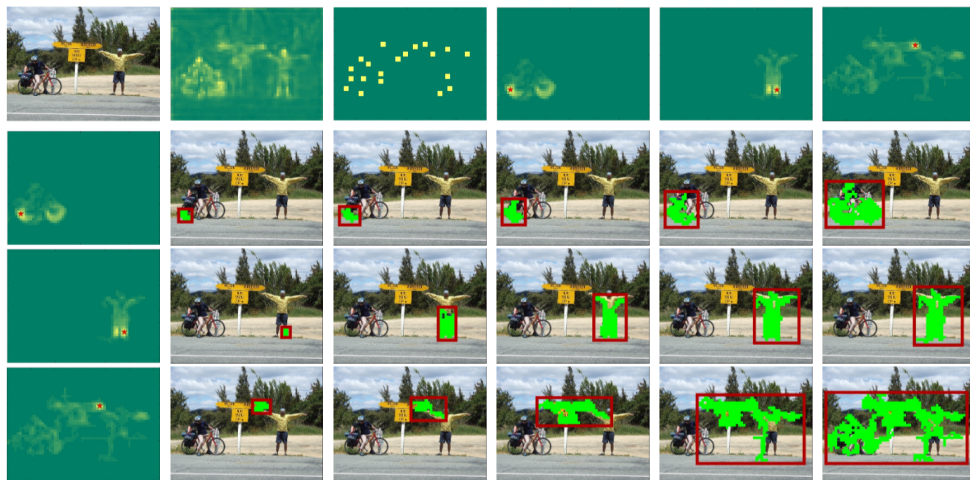


Figure 3.7: The Vo et al. method for Object Detection/Region Proposal. The top row depicts, from left to right, an input image, a summed CNN feature map, local maxima extracted from the map and three individual feature maps generated by calculating the dot product between the CNN feature vector at the position of a maxima and the feature map. Subsequent rows then depict these same three feature maps, followed by examples at different saliency thresholds of the main connected component generated from the maps being used to generate bounding boxes.

As can be seen in Figure 3.7, the authors provide a novel Region Proposal method based around image saliency. This method finds a set of local maxima within a summed feature map using persistence measurement [128], and for each maxima a new feature map is generated by creating a dot product between the original CNN feature map and the feature vector at the position of the maxima. The feature map produced from this dot product is then summed along

the filter axis much like before to get a new image, where the connected components algorithm is then applied with a bounding box around the component being the proposed region.

### 3.2.3 Applications in Waterborne Imagery

Object Detection and Region Proposal have become one of the most popular forms of Computer Vision for enhancing maritime navigation systems, specifically the use case of Maritime Surveillance [17, 112]. As the amount of global naval traffic increases and crews become smaller with the integration of smart systems, being able to identify nearby sea vessels allows for safe navigation as well as environment monitoring and anti-piracy measures.

However, traditional methods have mostly been developed for either urban locational imagery or object-based imagery, whereas waterborne imagery contain unique features and challenges including the tendency of water wakes to produce false positive detections [113], traditional assessment criteria for object detection not translating well to waterborne imagery due to non-linearity of physical distance between objects [129] and the diverse nature of imagery under various weather conditions.



Figure 3.8: Images taken from Figures in Bloisi et al.’s work [17], left image shows how existing object detectors cannot distinguish portions of land from boats and right image shows a low-resolution image of a boat with significant wake behind, which has been falsely detected as a second boat.

Bloisi et al. [17] proposed a maritime surveillance system that used a Haar based classifier to first detect a series of boats in an image, after which they remove false positives produced by boat-like shapes along the shore using a horizon line detector in order to find the separation between land and sea, common sense dictates that boats would not appear above sea level and as such these detections can be safely discarded.

As described in Figure 3.9, the Horizon Line Detector prevents false positive candidate lines by filtering out wakes and wave lines created by moving objects.

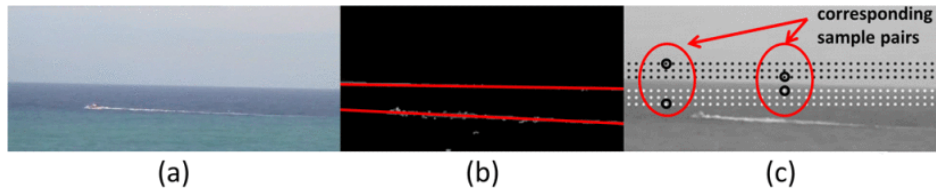


Figure 3.9: Horizon line detection method from Bloisi et al. [17], candidate lines are extracted from the input image (a) by applying a hough transform to its edge map, producing image (b). Candidates are validated by taking a rectangular set of sample points above and below the line, where corresponding pairs above and below are checked for differing intensities (See Image (c)), if 90% of pairs differ the line is considered valid.

In Boussetouane et al.’s work [112], a CNN object detection method was preferred over handcrafted, however, it was identified that state-of-the-art architectures such as Faster R-CNN are typically trained on small resolution images with large objects and minimal clutter [124, 126], whereas Waterborne Imagery tend to be high resolution with objects that appear small due to greater capture distance. As such, these pre-trained methods would not translate well to this type of imagery so the authors chose Fast R-CNN with HOG detection for region proposal as opposed to Selective Search.

The reasoning behind using simple HOG detection is due to Waterborne Imagery often being wide view and containing objects at various scales based on distance. HOG can efficiently analyse these images at various scales whereas the computation time of Selective Search grows exponentially relative to scale and image dimensions, the latter’s bottom-up nature also makes it perform poorly when tasked with identifying small objects in crowded scenes. A downside is that HOG can produce a noticeable amount of erroneous region proposals, however the authors deemed this to be acceptable as they could later be pruned by thresholding them by boat sub-class confidence scores.

More recently, Kim et al. [18] proposed the use of YOLO-V5 for maritime object detection on an augmented version of the Singapore Maritime Detection (SMD) [130] dataset called SMD-Plus. To alleviate the small size of the SMD dataset along with the large class-imbalance due to it consisting mostly of boats, the authors apply additional data augmentation techniques to their YOLO-V5 in the form of Online Copy-and-Paste and Mix-up techniques.

Copy-and-Paste consists of taking two individual sets of training images,  $\{I_1, I_2, I_3, I_4\}$  and  $\{J_1, J_2, J_3, J_4\}$ , adding color jitter via random alteration of brightness, hue and saturation components, copying objects from other training images and pasting them into the jittered  $\{I_1, I_2, I_3, I_4\}$  set, then, re-introducing set  $\{J_1, J_2, J_3, J_4\}$ , random mosaics are applied to both

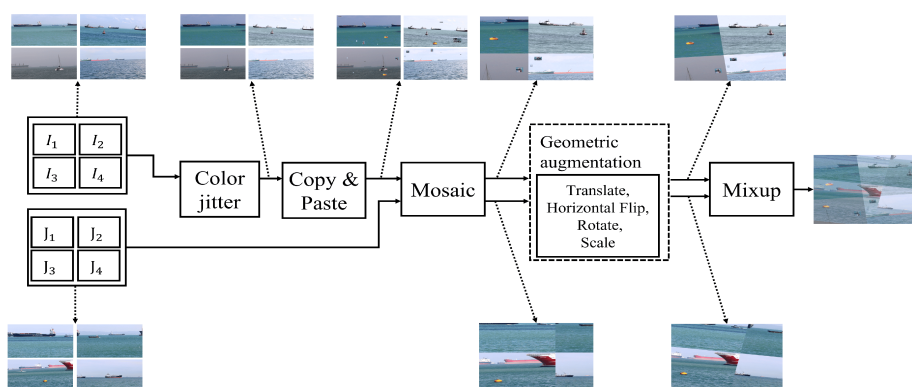


Figure 3.10: The Copy-and-Paste and Mix-up technique pipeline used by Kime et al. [18] on images from the Singapore Maritime Detection (SMD) dataset. These image augmentations are used to more effectively train YOLO-V5 on the SMD data.

it and the now augmented set, both output mosaics are then geometrically augmented before being “Mixed-Up” via a weighted linear interpolation of the two images and their labels [131].

### 3.3 Image Saliency

Image Saliency is a broad term that encompasses any area of computer vision research focused on detecting and highlighting local areas of interest in an image that the human visual system would be likely to extract from the overall scene, allowing the machine to mimic this behaviour to gain useful context from a single image [132]. In Explainable AI, saliency can also reflect the opposite of this, where the goal is to highlight which local areas of an image have been used by the AI to inform its decision making based on the global image features [133].

In both fields, image saliency is a useful tool for segmenting or promoting objects of interest within an image, separating them from less desirable background features and allowing us to save large amounts of computational resources and time by highlighting features of high importance, allowing developers to engineer their pipelines such that instead of having to process a whole image or multiple regions of an image to classify objects, areas highlighted by the saliency can be turned into region proposals to streamline the process and improve accuracy.

Before we continue, let us give a more proper definition of saliency. Saliency is a neurological term used to describe features that are prominent or “stand-out”, this can include visual objects of a particularly intense colour (i.e. red) in an otherwise dull looking scene or seeing something of significantly greater size than other features in an image [134].

Salient features generally come in two categories: Global and Local Feature Prominence. Global Feature Prominence describes saliency within the whole of an image, that is, taking an image as input and analyzing features across the whole image such as pixel contrast [135], what features are most prominent?

Local Feature Prominence on the other hand considers features of prominence across multiple components of an image, for example the Itti model [19], which we will soon cover in the following subsection, extracts features from a set of visual fields across an image which is also duplicated at differing scales to find salient features across scale and location in the image.

In either case, the most common output from image saliency is what's known as the saliency map, a 2D array usually matching the dimensions of the image it was generated from, where a continuous value between 0 and 1 represents the overall saliency of each pixel, these can then easily be thresholded to convert the map into a binary one instead.

### **3.3.1 Handcrafted Methods**

Because the motivation of Image Saliency was to mimic the Human Visual System (HVS), early methods were largely inspired by neuropsychology [136] with their being two primary models [137], bottom-up models that focus on low level visual features [138] and top-down models that use ground truth data to facilitate a task driven approach [139].

A classic example of a bottom-up approach is the Itti model [19] (Figure 3.11), which extracts intensity, colour and orientation values from an image in order to build feature maps that can be normalized and aggregated into a single saliency map in order to simulate the biological visual attention mechanism.

The Itti model calculates these feature maps at several image scales created with dyadic Gaussian pyramids in accordance to the center-surround operator of features, the architecture as a whole can be seen in Figure 3.11.

Within the architecture, we first produce colours, intensity and orientations maps. The intensity map is a simple average of RGB, whereas the colour map is a "colour double-opponent" system, where receptive fields are excited by one colour and inhibited by another, this is done by calculating color channel maps for red, blue green and yellow, then calculating new maps for opponencies that match those in the human primary visual cortex [140]. Finally, local orientation is obtained from the intensity map using angle components of the Gabor wavelet.

Using the various feature map scales, the model uses centre-surround differences between a finer, centre scale map and a coarser scale map for each of the three categories to obtain new

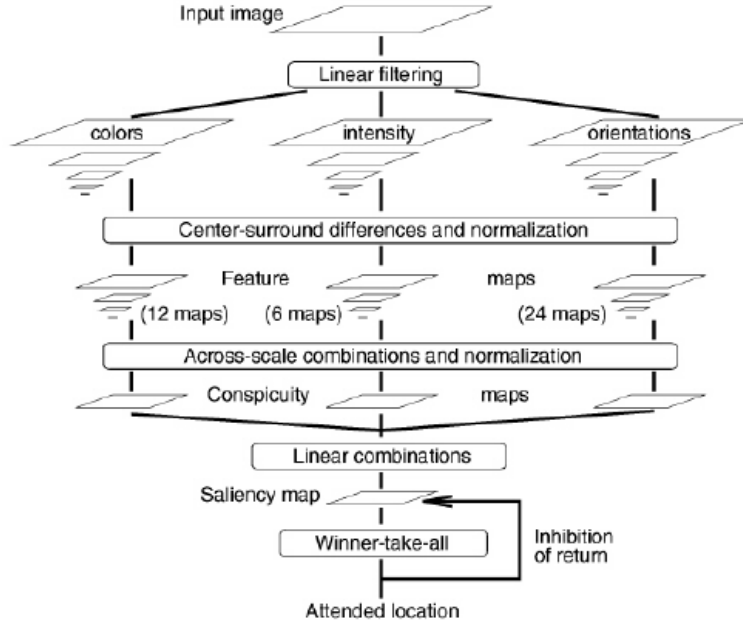


Figure 3.11: Itti model general architecture [19]

intermediate feature maps, which are then further processed by combining the various scaled maps and normalizing them before finally linearly combining them into a saliency map.

For the top-down approach, the Liu et al. [141] method is a well known example that focuses specifically on salient **object** detection as opposed to a more generalised visual attention representation. The type of object contained within the saliency is not considered to be particularly important, with training data coming in the form of bounding-box human-labelled images.

Given an image  $I$ , Liu’s approach calculates a binary saliency map  $A$  where for each pixel  $x$  in  $I$ , an equivalent pixel  $a_x$  in  $A$  has a value  $a_x \in \{0, 1\}$  where a value of 1 indicates that the pixels belong to a salient object. For a single image, the condition random field (CRF) framework is applied which involves calculating the probability of  $A$  given  $I$  as a conditional distribution  $P(A|I) = \frac{1}{Z} \exp(-E(A|I))$  where  $Z$  is a partition function and  $E(A|I)$  is the “energy” which is defined as so:

$$E(A|I) = \sum_x \sum_{k=1}^K \lambda_k F_k(a_x, I) + \sum_{x,x'} S(a_x, a_{x'}, I) \quad (3.2)$$

Here,  $F_k(a_x, I)$  represents the likelihood that pixel  $x$  belongs to a salient object with  $k$  be-



longing to a set of  $K$  unary features,  $S(a_x, a_{x'}, I)$  represents potential pairwise feature between  $x$  and a neighbouring pixel  $x'$ . The method learns to optimize the weights of feature set  $K$ ,  $\lambda_k$ , using maximized likelihood.

### 3.3.2 Deep Learning Methods

#### 3.3.2.1 General Methods

As with most fields of Computer Vision, the advent of Deep-Learning has presented new, more effective methods of performing image saliency detection, namely with the CNN and FCN models. Feature maps generated by CNNs pre-trained on image classification tasks such as ImageNet [124] can be seen as already having the inherent ability to present generalized image saliency without needing additional methods, training or prior-knowledge.

A good example of this is the work of Li et al. [20] (Figure 3.12) which uses multi-scale Deep CNN features combined with handcrafted to produce effective saliency maps. Using a CNN trained for ImageNet classification task, a saliency pipeline is built where given an image for which we want to calculate the saliency of a particular region, we take three nested regions where the smallest is the initial region of interest, the second are its surrounding neighbours and the third is the whole image.

Deep features generated from these are passed to an initial fully-connected layer for vectorization before concatenating these features and passing them to another fully-connected layer, this is then used to form the Deep Contrast Feature vector which is further concatenated with low-level handcrafted features to boost performance. Using ground-truth pixel-wise labellings of classified objects the models fully connected layers can be trained end-to-end.

Simonyan et al. [142] propose a method that generates class-based saliency maps, ranking pixels based on their influence on the class score, in other words the output loss function. As identified in their paper, the class score  $S_c$  for an image  $I$  could be represented as the following formula:

$$S_c(I) = w_c^T I + b_c \quad (3.3)$$

In this case, given that the result of applying weights  $w_c^T$  to  $I$  in results in a class score  $S_c(I)$ , we can assume that the magnitude of these weights reflect the importance of their corresponding pixels towards the output.

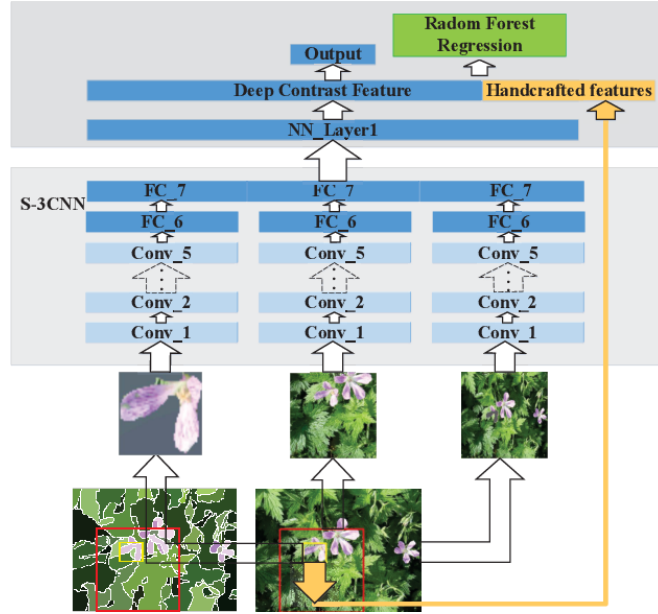


Figure 3.12: Li et al. [20] architecture for visual saliency detection using multiscale deep CNN features.

Unfortunately as identified in [142] the output function of a CNN is non-linear as opposed to the above formula, so in order to calculate per-pixel importance using weights  $w$  the proposal made by Simonyan et al. was to instead approximate  $S_c(I)$  by computing the first-order Taylor expansion:

$$S_c(I) \approx w^T I + b \quad (3.4)$$

Where, as stated in Simonyan et al.'s work:  $w$  is the derivative of  $S_c$  with respect to the image  $I$  at the point (image)  $I_0$ :

$$w = \left. \frac{\partial S_c}{\partial I} \right|_{I_0}. \quad (3.5)$$

A more recent CNN-based method is BASNet [21], which focuses on outputting high-quality boundaries that minimize the Intersection over Union (IoU) value between a boundary built around an output salient object and a ground truth box, in addition to the overall accuracy of the saliency map compared to a pixel-wise ground truth.

This is done by a U-Net like Encoder-Decoder network [32] which uses two CNN architectures, the Encoder CNN which acts as a regular CNN producing high-level feature maps from low-level input image features and the Decoder which takes the Encoder output and feeds them



preceded to digitally remove it, we would expect that score to drop to almost zero. We can then map these results to the pixels that were removed to indicate that they had a large effect on the class score for “dog”, producing a saliency map.

Methods that are able to achieve this goal in a generalizable manner include Mask [22], which learns image-specific perturbation masks that are able to reduce class score as much as possible with as minimal overall perturbation as possible; an example of this can be seen in Figure 3.14.



Figure 3.14: Taken from the Mask paper [22], we can see that by blurring the flute object within the image (i.e. Perturbing the image), we can reduce it’s class score for flute from 0.9973 to 0.0007, from which we can learn a “mask” which acts as a saliency map indicating the importance of the object to classification.

Mask achieves this by carrying out two optimization “games”, deletion and preservation; The deletion game consists of learning the smallest subset of an image, in the form of a deletion mask  $m$ , that causes the classification score of a desired object class  $c$  within an image to drop significantly, this is defined in their work as so, where  $f_c$  is the classification function for class  $c$  and  $x_0$  is the input image:

$$m^* = \underset{m \in [0,1]^\wedge}{\operatorname{argmin}} \lambda \|1 - m\|_1 + f_c(\Phi(x_0; m)) \quad (3.6)$$

The preservation game simultaneously learns the smallest mask subset that can retain the classification score, and is defined as so:

$$m^* = \underset{m}{\operatorname{argmin}} \lambda \|m\|_1 - f_c(\Phi(x_0; m)) \quad (3.7)$$

Another method is RISE [23] (Figure 3.15), which uses monte-carlo sampling to produce smaller binary masks that are then upscaled to generate  $N$  masks  $M_i$  of values within  $[0, 1]$  for an image  $I$ . By upscaling smaller masks, results in smoother masks that cover large portions of an image rather than masking pixel-wise, as each pixel can influence model score so more masks would be needed for good estimation.

$M_i$  are then multiplied against  $I$  to produce a new set of images  $I \odot M_i$ . Each member of this set is passed through the same deep model to produce a set of model output scores.

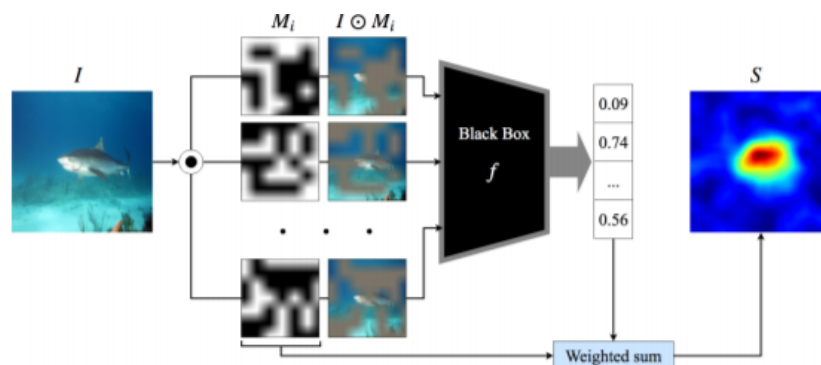


Figure 3.15: RISE [23] architecture for generating explainable saliency via linear combinations of loss based on a list of masked perturbations of the image input.

By multiplying the binary masks  $M_i$  by their respective output scores, we produce weighted masks that can be summed in order to create a saliency map describing pixel-wise importance towards model score.

### 3.3.2.3 CAM-based Methods

The basis of Class Activation Map (CAM) methods is the recognition that a saliency map could be generated using a linear combination of a global average pooled final convolutional layer [24].

This can be seen in Figure 3.16, where the classification weights generated from the global average pool are mapped back and multiplied by their corresponding activation maps, these are then linearly combined into a saliency map, in this case the score system was for an image classifier hence the map represents the salience of pixels towards the classification of “Australian Terrier”.

Grad-CAM [25] (Figure 3.17) was later proposed in order to allow a wider-range of models using fully-connected layers (i.e. VGG) to make use of the CAM-based approach, it does this by allowing an input image to be passed through CNN layers much like before and up to any task-specific output layer, then, setting the gradients for a desired object class to 1 and all others to 0, we backpropagate through the rectified convolutional feature maps of interest to combine them into the visual explanation.

A method that combines the ideas of perturbation and CAM-based approaches is Score-

### 3. Computer Vision Task Backgrounds and Applications for Waterborne Imagery

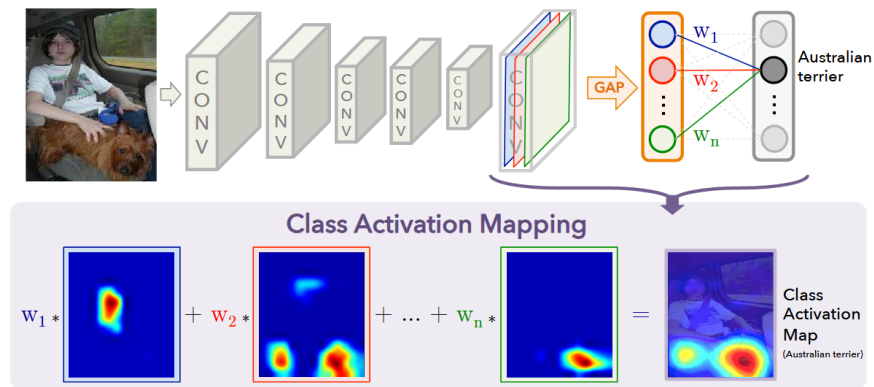


Figure 3.16: CAM saliency [24]. Global average pooling weights are mapped back and multiplied by their activation maps, which are linearly combined into the saliency map.

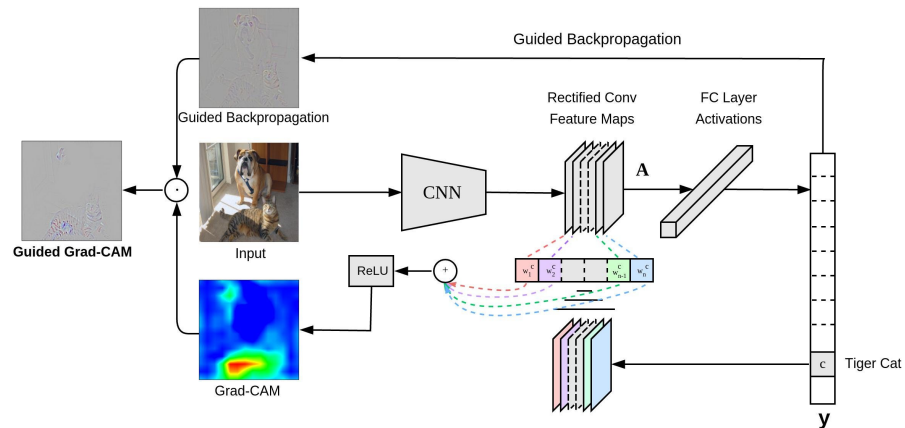


Figure 3.17: The Grad-CAM [25] architecture, here the desired output of the classification task is backpropagated to the rectified convolutional feature maps of interest, which get combined to produce the Grad-CAM heatmap. As seen in the figure, Grad-CAM also supports a guided variant by multiplying the heatmap with the networks guided backpropagation.

CAM [26], which seeks an alternative approach to generalizing CAM saliency maps in the form of using upsampled activation maps from the CNNs last Conv layer as perturbation masks to be used on the input image which are then fed through the CNN to generate class scores for each mask similar to RISE [23].

Just like with RISE, the scores for each are then mapped back and the final saliency map is a linear combination of these masks.

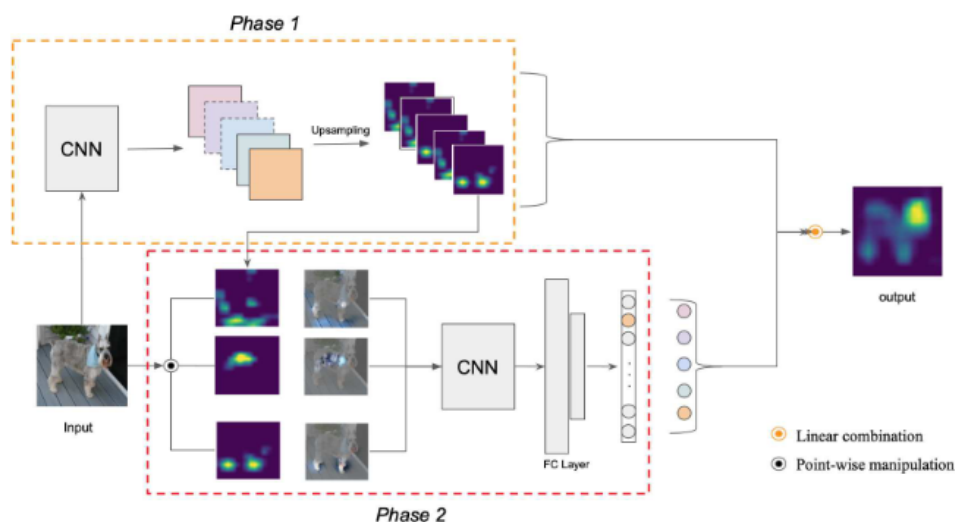


Figure 3.18: Score-CAM architecture from Wang et al. [26]. Initial activation maps are up-sampled to produce  $M_i$  masks for an image  $I$ , after which it essentially mimics the RISE architecture, multiplying the masks by  $I$  to produce  $I \odot M_i$ , each produces a score which can be mapped back to the activation maps and these can be summed to produce saliency.

### 3.3.3 Applications in Waterborne Imagery

Given that waterborne imagery are often sparse in terms of notable features, using image saliency to detect areas/objects of interest is useful for reducing the computational needs of onboard systems by allowing complex image algorithms to not waste resources on non-salient features [27].

Albrecht et al. [27] propose a Visual Maritime Attention framework which uses a saliency detector to find generic maritime objects, it does this through a combination of density, dissimilarity and surrounding features with an additional detector for sea and sky background (Figure 3.19).

In Sobral et al. [28] there is also a focus on separating maritime objects from the background (Background Subtraction), here using double-constrained Robust Principal Component Analysis (RPCA) on top of a saliency map.

Here, the saliency map is generated via a Boolean Map method [29] which is then used to produce to inputs to RPCA, the first being a normalized saliency map called the object confidence map and the second being a thresholded version of the map called the shape constraint, the final foreground mask is then generated via thresholding the RPCA sparse component.

This helps to tackle the challenge in maritime imagery of water wakes and waves acting

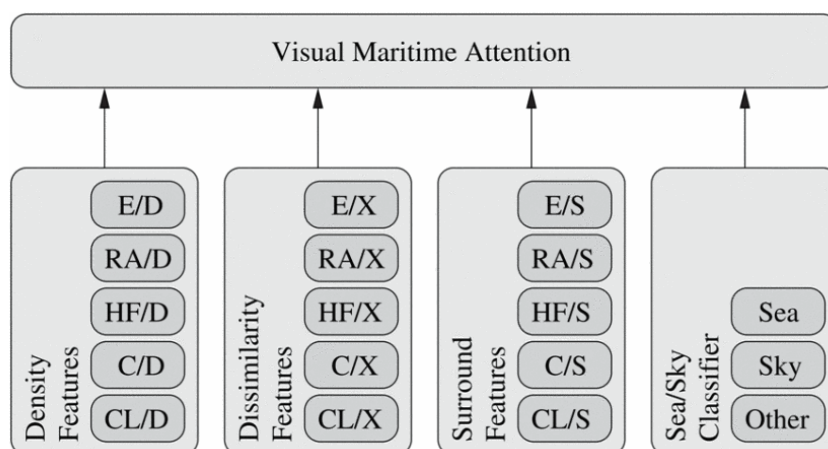


Figure 3.19: The Visual Maritime Attention framework [27], which uses associated edge ( $E$ ), right angle ( $RA$ ), high frequency ( $HF$ ), contrast ( $C$ ) and colour ( $CL$ ) metrics to calculate density ( $D$ ), dissimilarity ( $X$ ) and surround features ( $S$ ). Additionally, a sea and sky detector which takes in a HSV-Colour version of the image input is used, producing an 18-channel histogram with  $20^\circ$  separation, which is trained via Naive Bayes Classifier. All four features are aggregated and Naive Bayes Classification is used to get the final mapping.

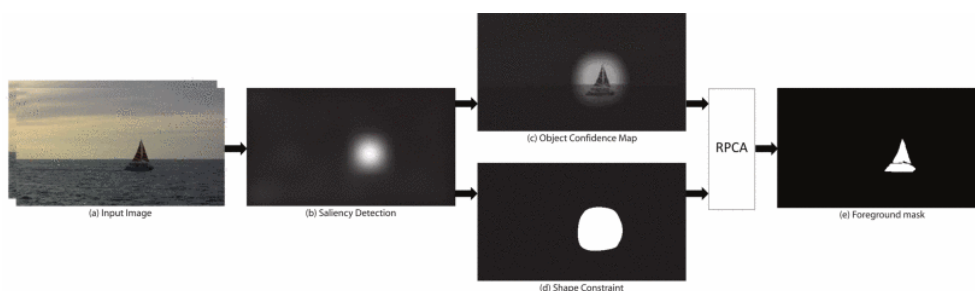


Figure 3.20: By using an initial saliency map to inform the generation of both a confidence map and shape constraint, RPCA can be applied on top of these to produce a more refined segmentation for maritime imagery [28].

as distractors for traditional saliency methods, such as Boolean Map method used standalone, which was shown in Sobral et al.'s work to be distracted by such features:

On the first row of Figure 3.21, we can see that BMS saliency is distracted by the wake left behind by the surfer on the left, whereas the RPCA method is more resistant to this. Second row shows small but visible distractions in BMS across the water beneath perhaps due to shimmer which RPCA is unaffected by, the final row also shows a distraction in BMS caused by the approaching wave.

In other applications, for images centered around objects or sub-images built using generic



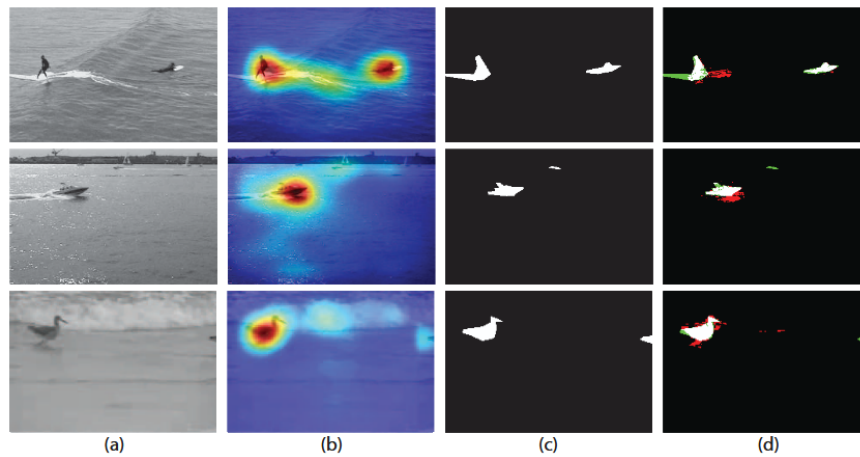


Figure 3.21: From left to right: (a) shows three input images, (b) shows BMS saliency map [29] (c) shows Ground-truth and (d) shows the RPCA method [28].

object and foreground detectors, we may want to apply image classification to label them more specifically.

One work that uses this approach is Xiong et al. [143], where the task is to explain the results of ship classifications using visual saliency, the researchers build a novel explainable attention network that takes remote sensing input images from a bird's eye view as input. The explainable attention network is built using two main components, the Causal Multi-headed Attention Module (CMAM) and the Filter Aggregation Mechanism (FAM).

The CMAM takes a feature map generated from CNN backbone as input and applies additional convolution to get high-level feature “attention” maps, after which a causal graph is used to describe the relationship between the input image, attention map and predicted labels. After the CMAM maps are multiplied by the CNN feature maps to create a new representation over which the original CNN features are then summed over, the FAM is then applied to make the resulting convolutional filters more explainable by filtering groups of filters based on distinct object regions that they represent.

For imagery that falls within our domain of being captured from a vessel, Baesens et al. [30] explore the use of Grad-CAM [25] for explaining Royal Navy ship classification on images from the Wright and Logan photographic collection within the National Museum of the Royal Navy, which had been vigorously labelled manually to include the name of the ship and the type according to standard navy classifications (i.e. Destroyers, Battleships, Carriers etc.).

It was found that Grad-CAM saliency maps had great potential as part of a decision support system for ship classification for curators and archivists, as the interpretable visual explanations

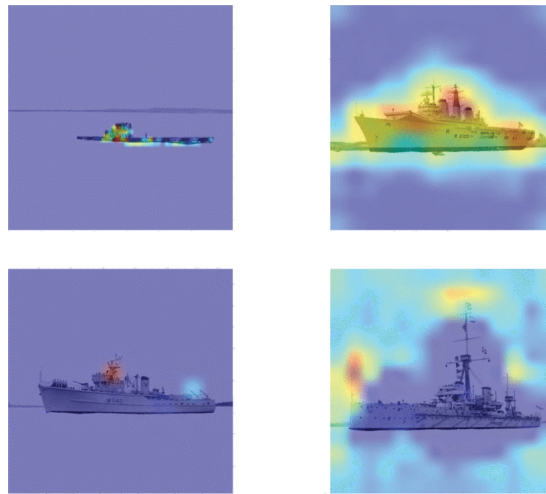


Figure 3.22: Grad-CAM [25] outputs for a CNN classifier trained on the Wright and Logan photographic collection within the National Museum of the Royal Navy [30]. In the top left, we see that Grad-CAM successfully highlights the sail of the submarine, top right the aircraft carriers ramp is highlighted, bottom left the minesweepers specialized equipment is highlighted and the bottom right describes a large edge detection filter around the large dreadnought.

were able to align with the experts analysis of what components were indicative of certain vessels.

### 3.4 Semantic Segmentation

Semantic Segmentation is a Deep Learning based Computer Vision task where the goal is to output a segmentation map where each pixel is assigned a specific class label based on some input image, as such it can be seen as an extension of image classification where in addition to labelling we also aim to localize objects in a scene. This makes it a challenging task as achieving a high score versus ground truth requires not only correct labellings but the localization of these labellings must have pixel-level precision in order to minimize loss.

When done correctly, semantic segmentation provides an even stronger knowledge-prior than most other Computer Vision tasks, as the segmentation map provides us with all of the information that image classification and object detection otherwise would (i.e. objects, labels and bounding boxes) with the added benefit of having all of this knowledge at pixel level precision.

There are some early methods that do not depend on Deep Learning which require substantial training data including supervised random forests [144] and visual grammar [145],

however most of the best performing methods would come later in the form of Deep Learning methods, which to this day maintain state-of-the-art performance on all publicly available semantic segmentation datasets [146].

### 3.4.1 Fully Convolutional Networks

One of the first major successes in Deep Learning-based semantic segmentation was the Fully Convolutional Network (FCN) proposed by Long et al. [31] (Figure 3.23), which replaced the dense layers of a traditional CNN with convolutional layers so that instead of outputting a 1-dimensional vector it would now output a 3-dimensional feature map. The FCN is considered a cornerstone of Deep Learning-based Semantic Segmentation as it pioneered the ability to modify CNNs in order to output dense predictions for semantic segmentation [147].

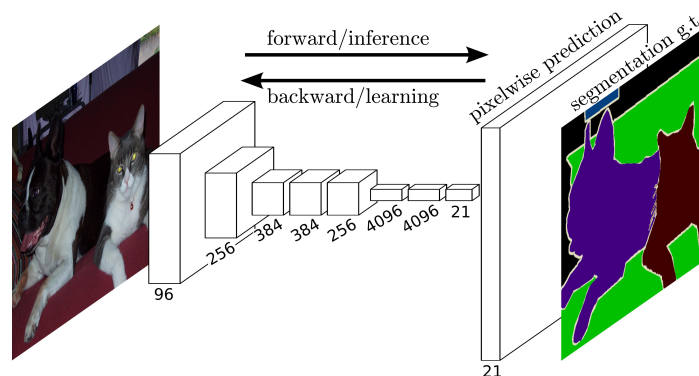


Figure 3.23: The FCN [31] uses convolutional output layers to produce a dense pixel-wise prediction map in a supervised end-to-end architecture.

In order to make the FCN work by connecting the output layer back to the individual input image pixels, the authors make use of interpolation based on the positions of input and output cells by introducing deconvolutional layers, which upsamples a given input by a factor of  $f$ , where  $f$  is treated as a fractional input stride  $1/f$ .

Finally, the FCN makes use of skip connections between its earlier layers and later ones, fusing the high-level semantic information learnt from the layers of the original classifier and the appearance information learnt by the additional deconvolutional layers.

Later, Ronneberger et al. [32] would propose the U-net which is based on a modified and extended FCN, here we now make use of two symmetrical yet opposite paths. The contractive path which acts like an ordinary CNN architecture with each “step” applying two  $3 \times 3$  convolutional layers to double the feature channels before applying a  $2 \times 2$  maxpool, the final

### 3. Computer Vision Task Backgrounds and Applications for Waterborne Imagery

output being a dense feature map not unlike the FCN, the expansive path is then made up of steps that apply two  $3 \times 3$  convolutional filters that halve the feature channels followed by  $2 \times 2$  deconvolutional layers for upsampling.

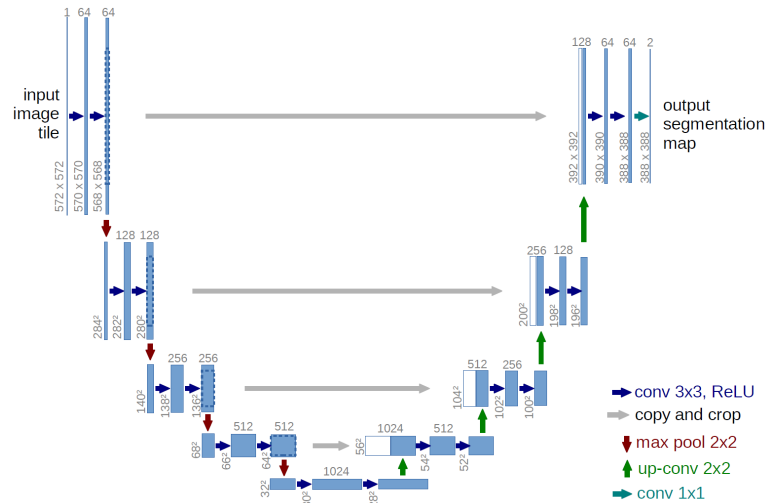


Figure 3.24: The U-Net architecture [32] uses a symmetrical contractive/expansive fully convolutional model in order to produce high-level semantic feature maps that can then be gradually upsampled into classified spatial features. Between corresponding convolutions on either side are skip connections, which re-introduce the semantic knowledge gained from the contractive stage to the expansive one.

As can be seen in Figure 3.24, between the convolutional layers of each step are skip connections once again inspired by FCN, which allow the output of the  $3 \times 3$  convolutional layer from the contractive path to be appended to the first convolutional layer of the expansive path.

Generally speaking, most Deep Semantic Segmentation networks are based on some form of FCN, with other examples including DeconvNet [148], SegNet [149], PSPNet [35] and Deeplab [150].

A more recent model from 2020 is the Parallel FCN [33], which aims to improve the low resolution output of FCN by introducing an additional branch to the architecture in the form of a holistically-nested edge detection network, which the authors define as a HED.

The HED in Ji et al.'s paper is based on the VGG network with fully connected layer and final maxpool layer removed, images are passed through the remaining layers with multiple "side-outputs" being retrieved at different points along the architecture, namely outputs from layers conv1\_2, conv2\_2, conv3\_3, conv4\_3, and conv5\_3.

Each side output effectively acts as an edge detection map at multiple image scales, these

### 3. Computer Vision Task Backgrounds and Applications for Waterborne Imagery

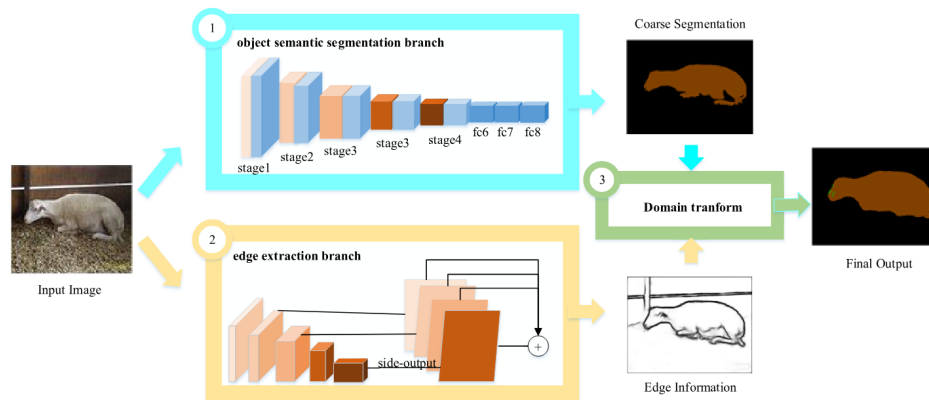


Figure 3.25: The Parallel FCN architecture from [33], the object segmentation branch is equivalent to a normal FCN, the edge extraction branch uses multiple VGG convolutional side outputs to fuse into an edge detection map. Outputs from these branches are then passed through a domain transform for a more refined output

are then fused using a hybrid layer to fuse the edge maps into the final result.

With this in mind, the Parallel FCN architecture passes an input image through both the regular FCN architecture [31] as well as the HED, producing an initial segmentation alongside an edge detection map which are used to create a final output through a domain transform.

#### 3.4.2 Deconvolutional Networks

Although deconvolutional layers are involved in almost all semantic segmentation models, the original FCN only made use of it to connect dense feature maps to input pixels, with U-net making use of DeConv mostly for upsampling. Since then, models built around using DeConv as a key component for segmentation have been proposed, the first being DeconvNet [148] (See Figure 3.26).

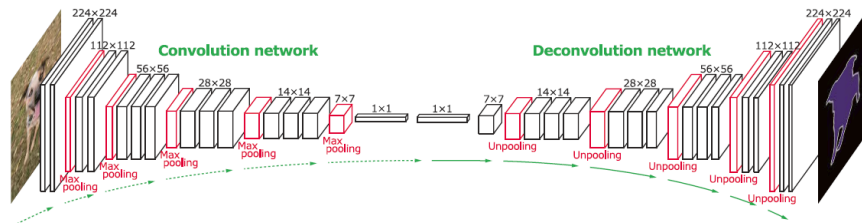


Figure 3.26: DeconvNet [32] makes use of a standard VGG-based CNN to output a dense prediction output which is convolved and unpooled gradually to produce a semantic segmentation map.

Structure-wise, DeconvNet is not dissimilar to previous examples in that it follows the same

two stage pipeline, with one stage using standard VGG16 CNN layers and the other using up-sampling to produce the predicted segmentation map. However, there are a few key differences, first, instead of being fully convolutional, the first stage produces a dense classification output vector similar to more traditional architectures, second, this dense vector is upsampled using two methods: “unpooling” layers and DeConv layers.

The unpooling layers for DeconvNet are similar to the  $2 \times 2$  upsamples used by U-net, the difference being that here we use switch variables which are recorded during the maxpooling layers to store the locations that each value in the maxpool output were pooled from when the  $2 \times 2$  filter was applied to the layers input. These switch variables can be brought back in the second stage for unpooling, allowing the pooled values to be reassigned to their original position in stage one.

As a result of the unpooling process, the outputs from these layers are initially sparse in appearance, in order to make these feature maps more dense DeconvNet applies DeConv layers with learnt filters. As discussed with FCN, a DeConv layer works opposite of a Conv layer, so instead of producing a single value from summed and weighted neighbouring pixels, DeConv produces a dense neighbourhood from a single value. In DeconvNet the output feature map is cropped at the end to maintain the same resolution as the unpooling layer output. Multiple DeConv layers are applied for each unpooling layer, with lower layers capturing overall shape and higher ones encoding more class-specific details.

In order to make the most of these class-specific details within the pipeline, DeconvNet is built to be used for instance-level image segmentation, meaning for a given image we gather a set of bounding box object proposals, perform segmentation on each of these proposal individually and aggregate results for the final output.

Another method that makes heavy use of the unpooling layer concept from DeconvNet is SegNet [149], which proposed a familiar yet novel encoder-decoder architecture where the 13 Conv layer encoder was simply made up of the 13 initial Conv layers of VGG16, the output of which is the maxpool output after the 13th Conv layer, making SegNet a fully convolutional network once again. The decoder is identical in structure meaning each encoder Conv layer and maxpool has an equivalent in the decoder with the difference being that the latter uses unpooling as opposed to maxpooling. Both the encoder and decoders have individual trainable filter banks for their convolutional layers.

In order to produce the final segmentation prediction, SegNet uses a trainable soft-max classifier which works on each pixel independently, producing a K-dimensional image where

### 3. Computer Vision Task Backgrounds and Applications for Waterborne Imagery

$K$  is the number of classes. The class with the maximum score for each pixel is the output prediction.

Compared to two previously mentioned methods that have similar architecture, DeconvNet and U-net, the authors of SegNet point out that, compared to the former, a lack of a fully connected layer in SegNet drastically reduces the number of parameters that need to be trained end-to-end, and, compared to the latter, SegNet does not make use of layer outputs from the encoder stage for concatenation with decoder layers, U-Net also does not make use of switch variables for determining the position of unpooled values.

A newer model, Contextual Deconvolution Network (CDN) [34], further improves upon these models by introducing spatial and channel contextual modules into the typical decoding stages of a Deconvolutional Network.

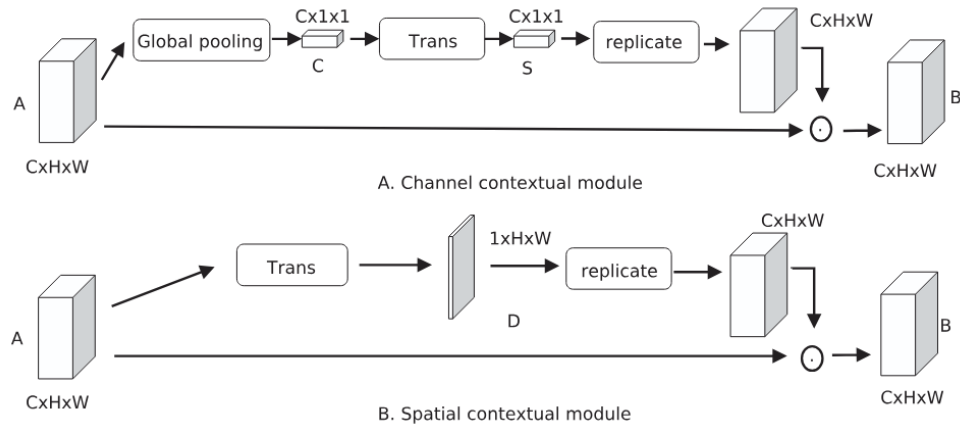


Figure 3.27: An illustration of the two contextual modules employed by CDN [34], (A) Channel Contextual Module and (B) Spatial Contextual Module

The aim of the channel contextual module is to retrieve contextual information across convolutional channel maps, given a feature map of dimensions  $H \times W$  and channels  $C$ , we first perform global pooling over  $H \times W$  to obtain a channel descriptor of dimension  $C \times 1 \times 1$ , this is then passed through a transform function which is similar to a regular convolutional module, applying a conv layer to reduce dimension, ReLU to learn nonlinear interactions between channels, an additional conv layer to increase dimensions and a sigmoid activation.

This produces a new channel descriptor  $S$  of dimension  $C \times 1 \times 1$ , containing contextual information between channels, this descriptor can be duplicated across the second and third dimension to create a feature map of shape  $C \times H \times W$  where every element across  $H \times W$  is equal to  $S$ , finally we multiply this by the original feature map to get a new channel context aware

feature map.

Spatial contextual module on the other hand is for gathering context of localized regions across a feature map, this module does not perform global pooling but instead passes the feature map to a different type of transform function, applying a conv layer with fewer filters for dimensionality reduction and using ReLU as before but here the authors now apply a conv layer with one filter to obtain a single channel map  $D$  of  $1 \times H \times W$  followed by a sigmoid activation.

This channel map is duplicated along the channel axis  $C$  times, so that we have a spatial context feature map of matching dimensions to the original input with which to multiply by and produce a spatially contextual feature map output.

These two are baked into the dense and compression units employed by Deconvolutional Nets, allowing the network to achieve superior performance [34].

### 3.4.3 ResNet-based Fully Convolutional Networks

So far, most of the segmentation methods we have covered make use of the VGG16 CNN architecture, however in addition to CNNs like VGG we also have access to more state-of-the-art image classification alternatives, namely the ResNet architecture [151] which upon release was able to overtake its competitors in the ImageNet Large Scale Visual Recognition Challenge, 2015.

ResNet or Residual Neural Network, is an architecture designed to combat the degradation problem in CNNs, which is the phenomena whereby adding too many layers will eventually cause performance on training data to saturate and degrade. It does this by using residual blocks (Figure 3.28) which save the coming input before passing it through multiple convolutional layers (two for ResNet), after which the output of these layers is aggregated with the initial input via skip connections. By stacking many residual blocks, ResNet can become very deep while achieving greater performance than models such as VGG.

A popular Semantic Segmentation that makes use of ResNet is PSPNet [35] or Pyramid Scene Parsing Network, which was motivated by a known issue of traditional FCNs in that they did not make sufficient use of global context priors for semantic segmentation. An example given in the PSPNet paper that illustrates this is an image of a boathouse with a visible boat, here the traditional FCN mislabels the boat as a car due to their similar features, however if the FCN was able to use the global knowledge that the image depicts a boathouse, it may have been able to more easily correct this mistake.

To remedy this, PSPNet, using ResNet with dilated convolutions [152] as a backbone to



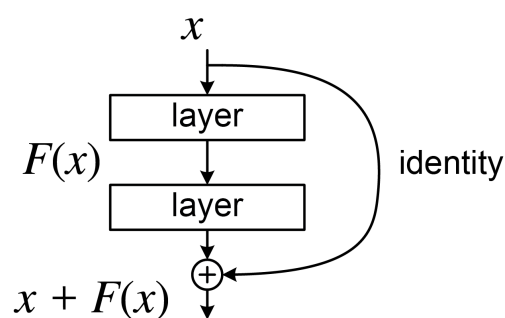


Figure 3.28: A residual block that takes an initial input  $x$ , passes it through various layers to get an output  $F(x)$  which is then aggregated to the original  $x$  via the skip connection (noted here as ‘identity’).

ensure greater receptive fields for global feature gathering, produces a feature map that has multiple different pooling layers applied to it in what is called a pyramid parsing module (Figure 3.29). Here, global context is gathered by pooling features of various sub-regions of the image with each pooling layer in the module determining their size and number.

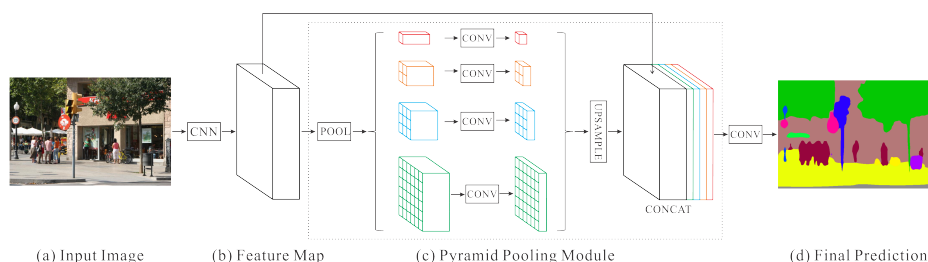


Figure 3.29: The PSPNet [35] architecture, which depicts the pyramid pooling module where outputs from ResNet CNN are passed through four pooling layers of increasing feature map size. These feature maps are convolved with a  $1 \times 1$  filter, upsampled and appended to the CNN output.

In order to facilitate deep supervised training for its ResNet-based FCN, PSPNet applies two losses to its ResNet101 backbone, the first being the main soft-max loss for training the classifier and the other being an auxiliary classifier loss applied after the 4th residual block. This auxiliary loss is allowed to pass through all previous layers along with the loss of the main branch, helping to optimize the learning process, the auxiliary is balanced via weighting.

Another state-of-the-art model that made use of a ResNet backbone was DeepLab [150], which also makes use of dilated convolutions (instead referred to as atrous convolutions here) in order to increase the convolutional receptive fields without an increase in parameters, returning an image size feature map via bilinear interpolation. DeepLab inserts these atrous convolution layers into the 4th residual block and uses them as part of its atrous spatial pyra-

### 3. Computer Vision Task Backgrounds and Applications for Waterborne Imagery

mid pooling (ASPP) scheme, which takes output feature maps from the ResNet backbone in order to efficiently record object features of various sizes.

Once this is done, the features generated from the pyramid of increasingly dilated (atrous) convolutions are applied resulting in equally shaped feature maps that are then convolved again using a  $1 \times 1$  filter so that their features can be summed before upsampling of the feature map as a whole.

Additionally, the authors of DeepLab propose applying a fully connected conditional random field (CRF) to the upsampled feature map in order to refine the segmentation result.

A more modern example is the Multi-path Residual Network [36], which was developed specifically to tackle high resolution polarimetric synthetic aperture radar (PolSAR) data, MP-ResNet uses the traditional FCN architecture with ResNet backbone for encoding image data.

The main contribution of MP-ResNet is that it attempts to maximize the visual receptive fields by using highly dilated convolutions while minimizing the loss of spatial information. PSPNet also achieved this through its use of the pyramid parsing module, which applied different global pooling algorithms to the same CNN output to retrieve convolutional features at different spatial scales, however the authors of MP-ResNet point out that achieving this effect through pooling is not as effective as stacked convolutional features.

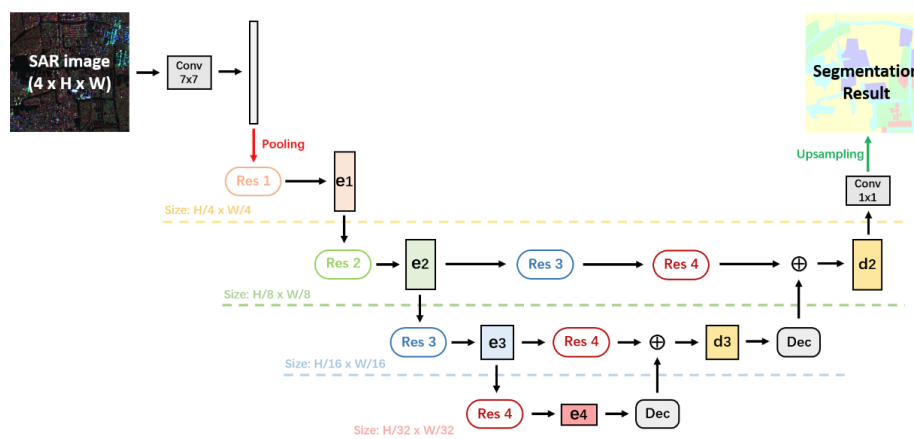


Figure 3.30: The MP-ResNet architecture [36], uses a standard FCN architecture with ResNet encoding, after encoding 2, 2 additional branches form applying ResNet modules 3 and 4 to the encoding 2, 3 and 4 in different orders in parallel. These outputs are then fused in the decoding phase.

To make use of stacked convolutional features, MP-ResNet uses two additional encoding branches after the second ResNet encoding block, these are arranged across the encoder network such that all outputs from that point on are passed through the same amount of con-

volutional layers. The outputs of each parallel branch are fused with one another during the decoding process, this can be seen in Figure 3.30.

#### **3.4.4 Applications in Waterborne Imagery**

Given the use cases for the previous Deep Learning methods of maritime surveillance, object detection and foreground-background separation, it is easy to see how semantic segmentation, a task that acts as a super-set of object detection and image saliency, could also be leveraged to gain a more spatially aware knowledge prior for waterborne image tasks.

In Cane et al. [113], deep semantic segmenters such as SegNet [149], ENet [153] and ESP-Net [154] were trained on a sub-set of the ADE20k [155] dataset containing relevant maritime based-imagery and ground truth labels such that they could then be applied to more relevant test images in order to segment objects within the images for the purpose of object detection.

Despite not having access to an ideal training dataset, having to significantly cut down on imagery from ADE20k due to it having a large focus on indoor scenery, all three methods showed promising results when trained on this custom dataset.

For a more waterborne-specific segmenter, Bovcon et al. [37] propose the Water Segmentation and Refinement Network (WaSR), which uses a novel encoder-decoder architecture, with the encoder generating deep features that are fused with the decoder and an optional Inertial Measurement Unit (IMU) feature channel used to aid in the detection of the water-edge [156].

The IMU measurement encoder allows the model to use encoded inertial data to project the horizon onto the image itself to aid in detecting the precise water edge, which is particularly challenging for a convolutional encoder to detect alone as camera haze induced by weather conditions or water obstructing the camera can blur feature maps around the true water edge.

The encoder is based on the segmentation backbone from DeepLab [150] which applies atrous convolutions to ResNet101. The encoder uses the output of residual blocks 2, 3, 4 and 5 to leverage both the generalized, low resolution features from the later blocks with the more fine high resolution information of the earlier blocks. These features are then passed to the decoder where they are fused with information from the IMU encoder in order to refine the final segmentation.

By fusing the segmentation features with the IMU data, WaSR eliminates faulty pixel classifications below and above the horizon line even when noticeable wakes appear, which introduce unwanted contrasting features along the water, examples can be seen in Figure 3.32.

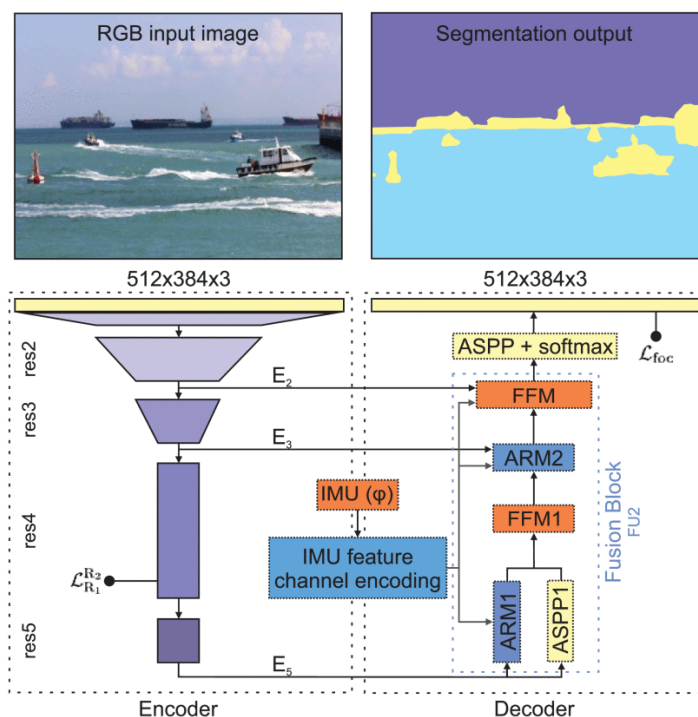


Figure 3.31: WaSR architecture [37], encoder generates high-level feature maps that are fused with the decoder, optional IMU feature channel provides assistance in detecting the visible water edge for more accurate segmentation.

### 3.5 Human-Centered AI

Throughout this chapter, we have covered three fields of Computer Vision, Object Detection, Image Saliency and Semantic Segmentation as methods for automatically identifying objects of interest, highlighting areas of importance and classifying image content respectively.

However, when we make use of such methods we are also taking away control from human users who, in the past, would have had to manually carry out said tasks themselves. This is not necessarily a problem as long as the AI is making correct predictions at the same level or greater than a human. In fact, for certain tasks, AI has overtaken humans in places such as the image classification competition of ImageNet [157].

For other cases however, the AI can still make mistakes that otherwise could be avoided through human intervention. This sentiment has recently motivated a growing community of research known as Human-Centered AI, which aims to preserve human agency through improving user understanding and control over AI tools [158].

To improve human agency, Human-Centered AI aims to always put the human first, such

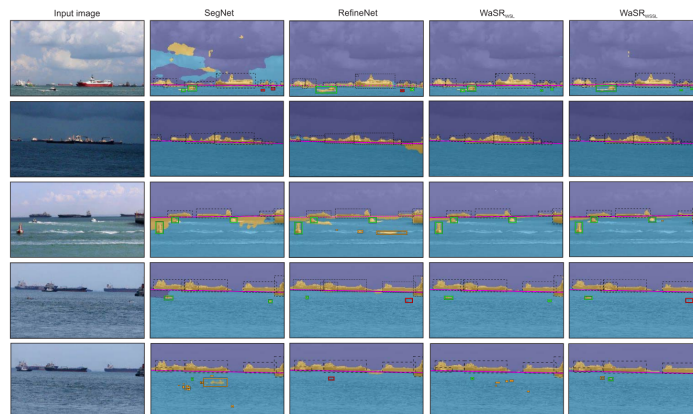


Figure 3.32: Each row depicts an input image and multiple segmentation maps output by several architectures for land (yellow), sea (light blue) and sky (dark blue). From left to right we have an input image, SegNet output, RefineNet output,  $WaSR_{WSL}$  and  $WaSR_{WSSL}$ . In columns 2 and 3 there are visible errors within the outputs highlighting sky as sea or sea as land pixels etc. due to wakes, reflections and weather conditions. Columns 4 and 5 show two sub-variants of WaSR, each of which is more resistant to these challenging conditions.

that it is always the AI that serves them [159], however there are some clear problems that need to be navigated in order for this to occur.

One of these problems is the black-box nature of AI which includes Computer Vision systems [160], when a user views the output of an Object Detection method for example, they have no way of knowing exactly how the model reached this conclusion, as researchers we can make some assumptions such as a CNN backbone identifying discriminative edge features but even so we cannot fully interpret the model's decision making.

Another issue is the previously mentioned lack of control, even when the user understands the AI's process they have no influence themselves on its decision making, the AI simply takes an input and gives an output, making them somewhat static and therefore prone to repeated error if not properly trained [161].

Here we will cover two Human-Centered AI approaches to tackle the above two issues; Explainable AI [38] and Human-in-the-loop [162], particularly in the context of Computer Vision.

### 3.5.1 Explainable AI

Explainable AI or XAI was a DARPA funding program made in 2017 [163], the goal of which was to build AI systems that could explain their rationale to human users, such that the user can

### 3. Computer Vision Task Backgrounds and Applications for Waterborne Imagery

understand how the AI behaves and predict it's future behaviour based on previous examples.

For Computer Vision, which usually consists of applying a Deep or Convolutional Neural Network to input imagery to detect or classify objects and categories, explanations should include why the model believes an object is present or why it classifies an object with a particular label [164].

In Xu et al. [38], two main branches of Explainable AI are defined; The first, Transparency Design, focuses on exposing the inner workings of an AI model directly to improve the developers understanding, this includes visually modelling the structure of the AI (i.e. Presenting the Neural Network node layout as a graph) and highlighting what intermediate outputs (i.e. Node activations or Convolutional Outputs) occurred before the models final decision.

The second is Post-Hoc Explanation, which focuses on presenting reasons as to the models output after it has been presented, to aid the end user. This can include textual descriptions of why a decision was made, visual explanations via Image Saliency or an explanation based on historical examples which show precedent for the models decision. A Figure depicting the use of both these branches is presented in Figure 3.33 from Xu et al.'s paper [38].

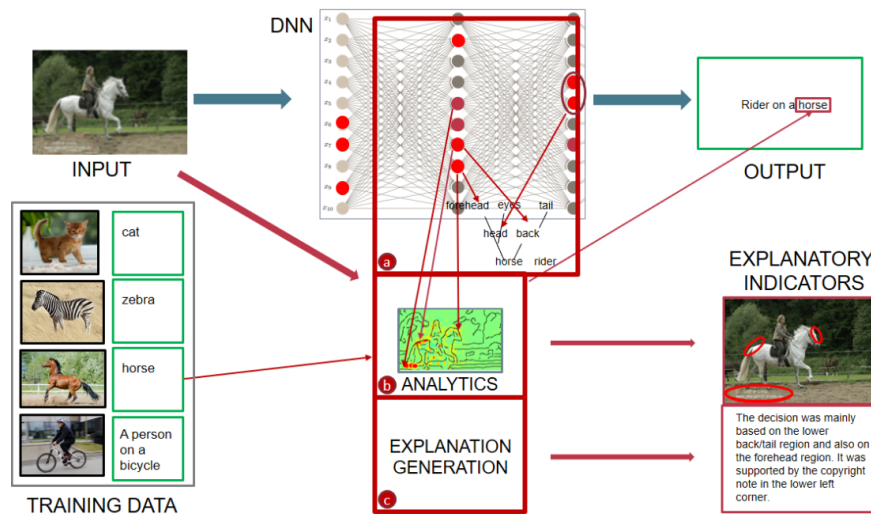


Figure 3.33: Figure taken from [38] depicting the two branches of Explainable AI at work. Boxes labelled a), b), and c), mark three ways in which a Neural Network classifier identifies an input as a “horse”. Box a), depicts a graph showing highly activated neurons that lead to the output decision which are mapped to the analysis in box b), indicating features that are often activated by these neurons to give the developer knowledge through Transparency Design. Box c), then shows how this information can be used to produce a Post-Hoc Explanation for the user.

In this work, I will later be focusing on end user experience with Deep VPR, as such the Post-Hoc Explanation branch of XAI is of particular interest. For image data, one of the most

common forms of explainable outputs is through the application of previously discussed Image Saliency methods.

Handcrafted image saliency were originally designed to detect notable content for enhancing model inputs, however later deep methods, particularly perturbation and CAM-based approaches such as RISE [23] and Grad-CAM [25], are mainly designed to tackle explainability, as their methodologies are performed post-hoc.

Figures 3.15, 3.17 and 3.18 in Section 3.3.2 of this thesis all show examples of how image classifiers can explain which features in an image contribute to their decision-making.

In Chapters 5 and 8, I will present Score-CAM [26], which combines aspects of RISE and CAM approaches, as a way of explaining Deep VPR outputs to users, this is enabled by the versatility of Score-CAM which is shared with RISE due to neither method needing a specific type of scoring algorithm to map feature importance, unlike Grad-CAM which was designed to work with Image Classification class scores.

### **3.5.2 Human-in-the-Loop**

For AI models, having access to prior knowledge allows them to make more effective predictions on individual test samples without the need to generalise on a large corpus of training data [165]. One of the best sources for prior knowledge are human experts, especially within specialised fields (i.e. Medical) which require years of training to correctly interpret useful information from [166].

Human-in-the-Loop is a system whereby a human provides input to an AI at some stage in the pipeline, a common method is to give a human user access to the predicted output for them to correct before passing it back through the model to promote optimization.

However, Human-in-the-Loop can take place outside of training, such as when a model produces an intermediate output before prediction, in which case the user can be shown this output, correct if necessary and pass it back to the model which can then produce an optimal output.

Human-in-the-Loop is closely tied with Human-Centered AI, as it gives the human user more direct control over an AI model. Examples include data integration and cleaning, where data gathered from multiple sources can either label identical objects using different terms or include errors in their formatting, both of which are easy for a humans to correct respectively but can have a negative impact on the AI if left unchecked. These methods both work to preemptively improve model accuracy by enriching the training data [167].

### *3. Computer Vision Task Backgrounds and Applications for Waterborne Imagery*

---

Another way in which humans interact with AI is through annotated ground truth data, which can often take more time to gather than it does to build the Deep Learning pipeline [168]. However, with modern AI making use of massive amounts of training data maintaining good quality annotations across all instances may simply not be scalable, an effective compromise then is to use Active Learning.

Active learning is a form of human in the loop where a pre-built algorithm or annotation based AI model is applied to initialize training data annotation, then based on some metric such as model uncertainty, particular examples are given to human users for re-annotation to hopefully optimize the model in terms of accuracy and classification certainty [169].

In Chapter 8, we will introduce a Human-in-the-Loop method for Deep VPR that allows users to view intermediate model outputs and edit them to remove errors for enhanced outputs, a form of Active Learning carried out in previous works [170].



## Chapter 4

# Creating the Plymouth Sound Dataset

### Contents

---

4.1	Data Collection . . . . .	<b>78</b>
4.1.1	Image Collection . . . . .	78
4.1.2	Locational Information Collection . . . . .	80
4.2	Dataset Features . . . . .	<b>81</b>
4.2.1	Perspective Changes . . . . .	83
4.2.2	Appearance Changes . . . . .	83
4.3	Availability . . . . .	<b>84</b>
4.4	Challenges . . . . .	<b>85</b>
4.4.1	Overabundance of Non-discriminative Features . . . . .	85
4.4.2	Obstructions . . . . .	87

---

## 4.1 Data Collection

In order to facilitate the work necessary to achieve our objectives, we required a waterborne image set suitable for Deep VPR, meaning we needed images taken from a vessel traversing some coastal shoreline and we needed multiple overlapping image sets taken at different times for train/test evaluation. At the beginning of our work, the only open source dataset that came close to our requirements was Symphony Lake [58], which depicted a bucolic environment meaning it was waterborne but was not wholly suitable for testing methods that we would wish to apply to maritime vessels.

Through collaboration with Marine AI Plymouth who provided assistance to our stakeholders of the UKHO, we were able to produce a simple 2D image dataset for Deep VPR covering the shoreline of Plymouth, UK. The resulting dataset, the Plymouth Sound Dataset (PSD), provides seven similar yet differing traversals around the area known as the Plymouth Sound.

### 4.1.1 Image Collection

Images were collected from the video feed of a multi-view camera system mounted on the IBM/Promare Mayflower Autonomous Ship at 20 second intervals with the vessel travelling at an average of 5-7 knots (2.5-3.6 m/ps). This means the density of image captures along the traversals are more sparse when compared to other datasets such as Symphony Lake or Tokyo 24/7.

This was decided based on the fact that, visually, the visible shoreline would not change much over shorter intervals given the large distances at play. With Plymouth Sound being roughly  $6km^2$ , once out of port the boat would often be more than  $1km$  from the nearby shorelines and landmarks.

Given this large area and relatively sparse capture rate, our radius of ground truth for each image is set to  $250m$  which is a large increase compared to the  $25m$  used by NetVLAD for Tokyo 24/7.

The multi-view camera system consists of six views with standard 2D image capture, including one pair of cameras looking in the direction of the ships port and starboard and two camera pairs looking out towards the bow and stern (See Right of Figure 4.1).

Images taken from cameras 0, 1, 2 and 3 have a resolution of  $1920 \times 1080$ , whereas camera 4 has a resolution of  $1024 \times 768$  and camera 5 has a resolution of  $1280 \times 720$ . As mentioned

#### 4. Creating the Plymouth Sound Dataset



Figure 4.1: Left: Image of the IBM/Promare Mayflower Autonomous Ship (MAS) which was used for image capture with its multi-view camera system mounted. Right: Simple Diagram of the arrangement of the multi-view camera system, each circle represents a single camera and ID.

previously, these were taken along seven similar but varying runs across the area of Plymouth Sound, UK, with some runs only going so far as Cawsand Bay and others as far as Eddystone Lighthouse in the English Channel south of Plymouth Sound:



Figure 4.2: Maps depicting the path of each run along the Plymouth Sound and beyond.

Before any post-processing is done, the total images contained for each run from top left to bottom right are 4807, 5518, 6532, 6572, 6792, 6827 and 6858 for a total of 43906 images. However, for a large period of time within these sets, the Mayflower Autonomous Ship would be docked in Turnchapel Wharf with the cameras still on, sometimes overnight, meaning there is a large spike in locational density around this location.

To remedy this, our post-processing uses the speed in knots field from the GNV TG data recordings (Outlined in Table 4.2), to determine which images were captured while the boat was moving outside of it's docking zone based on if their speed in knots was greater than 0.5.

#### 4. *Creating the Plymouth Sound Dataset*

---

From this group we took all samples, from those whose speed was less than 0.5 knots, we take only a random sample of 1% to still have a small group of dockside images. The total images for each run after this were 2760, 2268, 1380, 2382, 4224, 3066 and 593 for a total of 16673 images.

All images are stored in JSON files which include UNIX timestamp, color format, camera ID and the image itself in base64 encoding, filenames begin with the UNIX timestamp followed by an underscore followed by the camera ID.

##### **4.1.2 Locational Information Collection**

Locational Information was captured via the MAS's onboard systems independently of our image collection, all captured information is initially stored in a large text log containing rows of data belonging to various navigational data formats. There are four main GNSS-based formats contained within the logs indicated by headers at the start of each row, including GPGGA, GNVTG, GNHDT and GPDZA whose contents are described in Tables 4.1, 4.2, 4.3 and 4.4 respectively.

For Place Recognition, the only data we needed access to out of these is the Latitude/Longitude information from the GPGGA logs to evaluate the distance between query images and retrievals and thus determine success/failure. Knots values from GNVTG logs were also used for filtering data captured while the MAS was docked/not moving so that we could focus on images along the traversal only.

Before the log header for every row is a UNIX timestamp, these can be used to find matching/closest timestamped logs for each image in order to merge them with the relevant data.

#### 4. Creating the Plymouth Sound Dataset

---

Field	Data	Description	Example
1	\$GPGGA	Log Header	\$GPGGA
2	UTC	UTC Time	230000.00
3	Latitude	Latitude (DDmm.mm)	5021.62020
4	Latitude Direction	Latitude Direction	N
5	Longitude	Longitude (DDDmm.mm)	00406.95436
6	Longitude Direction	Longitude Direction	W
7	Quality	GNSS Quality	5
8	Satellite No.	Number of satellites in use	13
9	HDOP	Horizontal Dilution of Precision	0.9
10	Altitude	Antenna altitude above/-below mean sea level	0.78
11	Alt Units	Unit of measurement for Altitude (i.e. M for Metres)	M
12	Undulation	The relationship between the geoid and the WGS84 ellipsoid	52.60
13	Und Units	Unit of measurement for Undulation (i.e. M for Metres)	M
14	Age	Age of correction data in seconds	008
15	Station ID	Differential base station ID	0381

Table 4.1: GPGGA Data Format

## 4.2 Dataset Features

Being one of the first open source datasets that we know of covering a large distance water-borne area, images in our dataset present a host of unique features when compared against urban imagery. Generally speaking, with the exception of images taken around Turnchapel wharf which depict a more urban port environment, our images feature more sparse locational information compared to those taken on land, many images may even contain no features if the camera was pointed towards the sea when it was captured.

Most locational features appear just above the visible horizon, that being the nearby shore-

#### 4. Creating the Plymouth Sound Dataset

Field	Data	Description	Example
1	\$GNVTG	Log Header	\$GNVTG
2	Track True	Track made good, degrees True	210.812
3	T	True track indicator	T
4	Track Mag	Track made good, degrees Magnetic	210.812
5	M	Magnetic track indicator	M
6	Speed Kn	Speed over ground, knots	5.047
7	N	Nautical speed indicator (N = Knots)	N
8	Speed Km	Speed in kilometres/hour	9.346
9	K	Speed indicator (K = km/hr)	K
10	Mode Indicator	Positioning system mode indicator	D

Table 4.2: GNVTG Data Format

Field	Data	Description	Example
1	\$GNHDT	Log Header	\$GNHDT
2	Heading	Vessel heading in degrees	202.2537
3	T	Indication that degrees are True	T

Table 4.3: GNHDT Data Format

Field	Data	Description	Example
1	\$GPZDA	Log Header	\$GPZDA
2	UTC	UTC Time	230000.00
3	Day	Day in XX format	29
4	Month	Month in XX format	03
5	Year	Year in XXXX format	2022

Table 4.4: GPZDA Data Format

lines of Plymouth Sound, Rame and some of Whitsand Bay. Each run is taken on subsequent days to ensure variable weather conditions between image sets for more robust Place Recognition evaluation.

### 4.2.1 Perspective Changes

As mentioned previously, images were captured simultaneously from the multi-view camera system aboard the MAS at fixed intervals, meaning every image has a corresponding set it is associated with giving the user a view of all directions depicted in the right of Figure 4.1 at the time of capture.

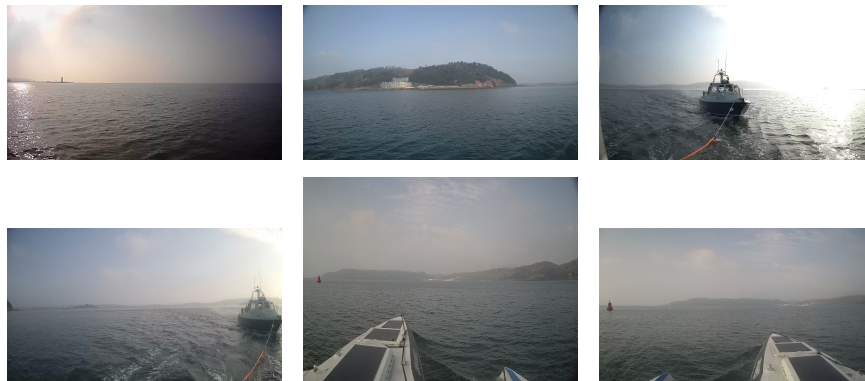


Figure 4.3: An image set from MAS’s multi-view camera system. Top row (left to right): Images from camera ID 0, 1 and 2. Bottom row (left to right): Images from camera ID 3, 4 and 5

This gives users the opportunity to either treat each image as an individual retrieval problem for Deep VPR or to use the set for each location. For simplicity sake, in this thesis we treat every image as an individual during training and testing stages, however as each image is labelled with the same UNIX timestamp they can easily be grouped together for multi-view image retrieval.

As each run has a different traversal path (See Figure 4.2), certain key locations are captured from alternative locations across the entire image set, introducing additional perspective changes on top of those produced by the vessel looping back to the starting position in each run.

### 4.2.2 Appearance Changes

In addition to changing perspectives between runs, different weather conditions on each day a run was captured on provide varied images of similar locations (and thus positions), for example see Figure 4.5 below where we show Fort Picklecombe, a notable building around the western shore of Plymouth Sound:

There are also more notable weather differences between particular runs, for example run 7 which is not depicted above, was captured during a heavy fog and as such has no/limited

#### 4. Creating the Plymouth Sound Dataset

---

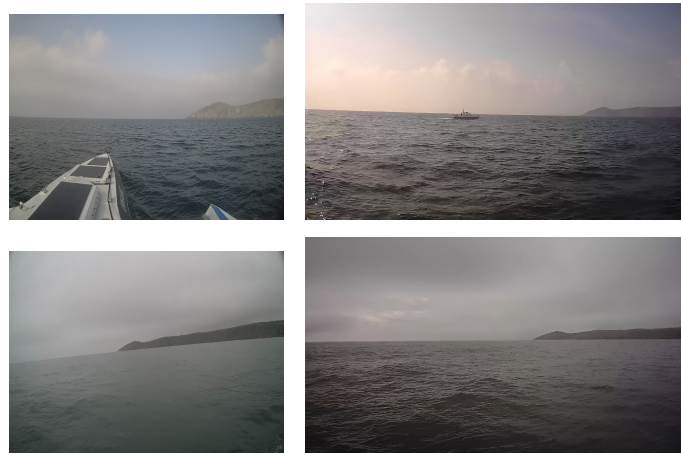


Figure 4.4: Top Row: Images of the Rame's Head taken from run 1 during the MAS's embarkment (left) versus disembarkment (right). Bottom Row: An additional embarkment/disembarkment image pair of the Rame's Head from run 4.

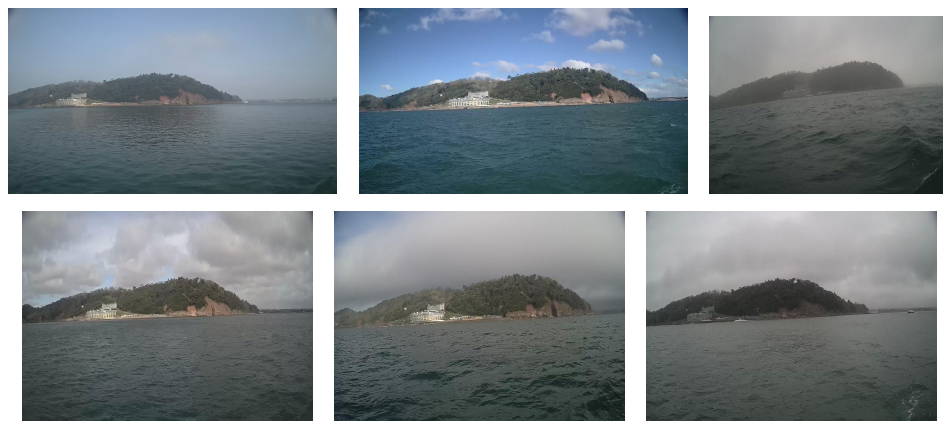


Figure 4.5: Similar views of Fort Picklecombe from runs 1-6, between these there are small differences in hue caused by cloud coverage as well as greater differences caused by fog such as in run 3 (Top Right).

visibility of shoreline features, this fog also informed the MAS to disembark earlier than other runs for safety reasons. On the flip-side, images from run 2 are relatively clear as the weather was clear that day.

### 4.3 Availability

The dataset has been made open-source by Marine AI Ltd. under the name “MSubsLtd-Mayflower-USV-Imagery” and can be found on Kaggle using the following link:



#### 4. Creating the Plymouth Sound Dataset

---

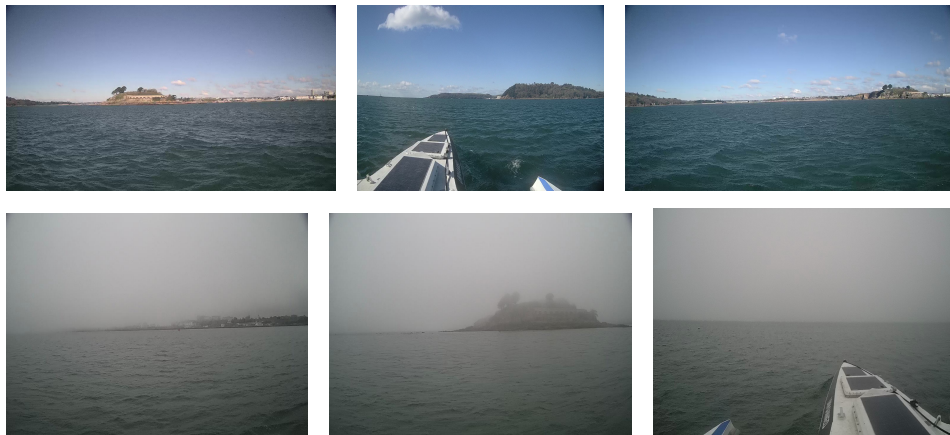


Figure 4.6: Top Row: Images from Run 2 (03-31-2022) which were taken during exceptionally clear weather. Bottom Row: Images from Run 7 (04-14-2022) which were taken during heavy fog.

<https://www.kaggle.com/datasets/marineailtd/msubsltd-mayflower-usv-imagery>, there are no copyrights associated with this repository.

It contains the data collected from the seven runs presented in this paper in folders labeled with the date of capture, within each folder are the images we produced after post-processing, along with a CSV file containing all locational, speed and heading data gathered from the data formats described in Section 4.1.2 for each image.

## 4.4 Challenges

### 4.4.1 Overabundance of Non-discriminative Features

Previously, we described the issue of sparse locational information (i.e. Visible shoreline), because of this we run into an additional inverse issue, an overabundance of non-informative information, sea and sky. It is not difficult to see that, as categories, “sea” and “sky” are not useful, discriminative features for visual place recognition.

In Figure 4.7, we can see that, after applying the WaSR segmenter to all images in our dataset, the average percentage of pixels belonging to land across all images is far lower than both sea and sky, with the last category being the most dominant.

Sea (Or “water”) as an image feature captured from a non-aerial position only serves to create a buffer zone between the bottom of the image and visible shoreline in the middle, the only locational information one can derive from the presence of water is that the image is likely taken from a vessel, but seen as this information is already known to us when considering the

#### 4. Creating the Plymouth Sound Dataset

---

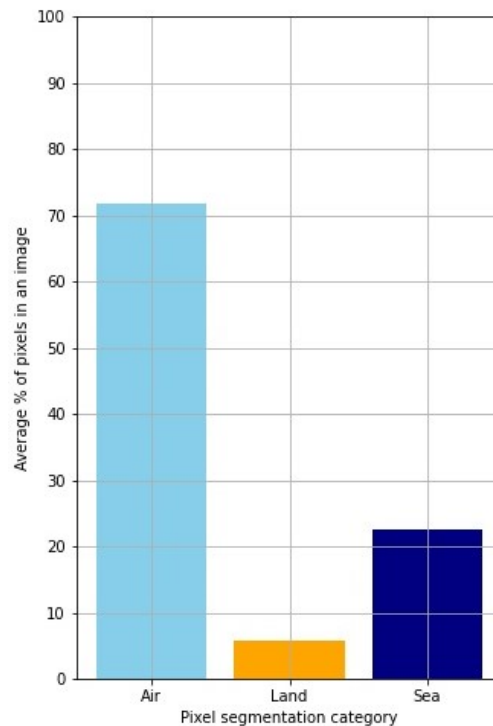


Figure 4.7: The average % of pixels within Plymouth Sound Dataset images belonging to the land, sea and sky categories defined by WaSR semantic segmentation.

Plymouth Sound dataset, it is of little to zero value.

This would only present a minor inconvenience if sea were a constant, invariant image feature, but this is not the case. Waves and fast winds can cause the “texture” of the sea to change in the image introducing natural variance between shoreline imagery, this can be seen in Figure 4.5. When a vessel moves across an image, or when taking imagery from behind the capture vessel, wakes are produced in the water which introduce new edge features along the water [113].

Sky is very similar, it tells us little to nothing about where we may be located geographically and clouds, fog and time of day greatly impact its visual composition, introducing variance between locational imagery, an example of this can be seen in Figure 4.6.

In Chapter 7 we will cover three solutions relating to this challenge, all of which are derived from using a semantic segmentation model for identifying sea, sky and land pixels, then leveraging this knowledge to promote features belonging to the land category, exclude regions containing minimal land pixels when extracting local feature descriptors for SSM-VPR and only analysing spatial features along the top edge line of the largest land pixel “object” ex-

tracted via semantic segmentation.

#### 4.4.2 Obstructions

In terms of obstructions, there are three main challenges, firstly, in images from the front-facing cameras of the MAS, the bow of the vessel becomes visible and can act as a distractor, this is the least problematic as we can simply crop out this bottom section of the image, which also reduces the amount of water.

Secondly, in images captured by cameras facing the stern, an observation vessel follows the MAS along its travel path, likely for safety reasons, this provides a consistent obstruction which is both a negative and a positive, as it opens opportunities for boat detection and erasure experiments.

Thirdly, the camera is obstructed in some runs, typically those in harsher weather, by a layer of water. This blurs the cameras view of the surroundings and makes features difficult to extract via CNN architectures due to the removal of distinctive edges.



Figure 4.8: Top Row: Images facing the bow of the MAS which acts as a distractor. Middle Row: Images facing the stern containing the observation boat accompanying the MAS. Bottom Row: Images from run 3 which had the camera obstructed by water.

Unfortunately, in this work, we did not have sufficient time to work on this challenge area, as such solutions to these challenges are left to future work.

## Chapter 5

# Evaluating Waterborne Deep VPR

### Contents

---

5.1	Introduction . . . . .	89
5.2	Proposed Approach . . . . .	90
5.3	Method . . . . .	90
5.3.1	Datasets . . . . .	91
5.3.2	Architecture . . . . .	93
5.4	Comparative Analysis and Results . . . . .	96
5.4.1	Quantitative Analysis . . . . .	98
5.4.2	Qualitative Analysis . . . . .	99
5.5	Summary . . . . .	100

---

## 5.1 Introduction

This work seeks to prove the viability of unmodified state-of-the-art CNN-based Deep VPR for waterborne imagery by comparing quantitative results from a set of land-based image sets to waterborne image sets. In addition, saliency maps generated by Score-CAM [26] for retrieved images are shown, which are calculated by substituting the typical classification loss used to weight salient features with the euclidean distance between query and retrieval image.

Place Recognition has been a topic of interest within the fields of robotics and autonomous cars for many years, however, with the emergence of state-of-the-art deep learning pipelines, previous benchmarks set by older hand-crafted methods have been vastly improved upon. This has helped to re-invigorate research into place recognition for autonomous navigation using new state-of-the-art Deep VPR models.

However, across most of this Deep VPR research the imagery used for training and testing is land-based or urban in nature, meaning they are typically captured from the point of view of a land vehicle, most commonly some type of car. Datasets belonging to this category include the KITTI dataset [171], the Berlin datasets [67] and Nordland [172].

Developing Deep VPR pipelines for the land domain is of course greatly important, but it is important to consider other potential domains also. More specifically, this work seeks to extend the field of Computer Vision to the waterborne domain, where autonomous navigation has also become a topic of great interest due to Deep Learning now offering what may be seen as a feasible implementation of such a system [42–44].

With this in mind, can current state-of-the-art approaches work on waterborne imagery? Most Deep VPR pipelines are almost unanimously built on top of a CNN backbone [5, 6, 73, 74, 76] such as VGG [69], in addition these tend to come with pre-trained weights which are most commonly learnt from ImageNet [124] or sometimes from a more location-based dataset such as Places365 [173].

For waterborne imagery, ImageNet weights are unlikely to translate as on the water individual objects are much more sparse than on land and the greater distances between camera and visible features means that geometry becomes more reliable for identification. Places365 provides a more useful initial backbone then, but even still there may be common structures and challenges within waterborne image scenes that this dataset does not train for.

By presenting novel saliency maps for the output retrievals, this work presents a more interpretable look into what features between the land and waterborne domains are valued by the Deep VPR model in order to inform further adaptations.

## 5.2 Proposed Approach

I propose a quantitative evaluation of a current state-of-the-art pre-trained CNN-based Deep VPR pipeline on two types of imagery, land-based imagery and waterborne imagery, with the latter being defined as imagery captured from a waterborne vessel. I propose the SSM-VPR model [5] as the chosen pipeline due to, as during 2019-2020, when this research began, this method was achieving state-of-the-art results on a variety of VPR datasets, outperforming previous state-of-the-art methods such as NetVLAD [6] and Region-VLAD [57].

Following this work, the SSM-VPR model has continued to be a highly effective and relevant Deep VPR approach, being cited in works such as Masone’s 2021 Deep VPR survey [82] who recognized the pipelines spatial matching method as a prime example of non-geometric re-ranking, Tsintotas’ 2022 survey on visual loop closure detection [174] and Sousa’s 2023 literature review on long-term localization and mapping [175], where SSM-VPR was still cited as an example of VPR models using convolutional layers achieving impressive benchmark results.

In addition, I propose a novel Deep VPR-based visual saliency output for suggested retrieval images in order to explore the application of traditionally classification-based explainable AI to the field of place recognition. The expected outcome of this is for salient features in a retrieved image to represent what features are shared between itself and the query, this should give us additional insight into the inherent differences between the two image domains by highlighting what features are most highly activated across each.

To achieve this novel salient output, I use the Score-CAM [26] method, both because it has been proved to achieve state-of-the-art results in the field of explainable AI when compared to its predecessor Grad-CAM [25], and also because it does not require gradients in order to work which is essential for use with SSM-VPR as the pipeline is technically unsupervised and as such gradient-based methods could not be applied.

## 5.3 Method

The proposed method consists of applying the SSM-VPR architecture to several test folds from both land and waterborne place recognition datasets, Precision-Recall curves are drawn for each fold and compared to see if SSM-VPR can achieve similar performance. Afterwards, qualitative analysis is performed using Score-CAM on the retrieved images via a novel computation to highlight areas of similarity and high activation.

### 5.3.1 Datasets

#### 5.3.1.1 Land-based Dataset

My chosen dataset for land-based imagery are the Berlin image sets which come in three folds representing individual locations within the city, that being Kudamm, Halenseeestrassen and the A100. For each fold a training and testing image set is provided which cover the same path through each location taken from different viewpoints. These provide three separate land-based image sets for PR-Curve comparison, each of which have a consistent style of image capture between them while also having a good deal of variety.



Figure 5.1: Examples of ground truth image pairs of the same within the three Berlin image sets. Top Row: Halenseeestrassen. Middle Row: Kudamm. Bottom row: A100.

For evaluation, the training images are stored within the Deep VPR search-space while the test images are used as image queries.

#### 5.3.1.2 Waterborne Dataset

At the time this work was carried out, the Plymouth Sound Dataset had not yet been captured due to COVID-related interference, as such I sought open source image sets suitable for Place Recognition evaluation, ‘suitable’ in this case being a dataset that contains images from a



## 5. Evaluating Waterborne Deep VPR

---

similar traversal over different time periods in order to evaluate the models ability to identify key locations under multiple perspectives and conditions.

The open source dataset that most closely matched this criteria while still being Waterborne was Symphony Lake [58], which contains a multitude of traversals along the banks of the associated lake in Metz, France. Although this dataset more specifically depicts a bucolic environment [176] as opposed to a shoreline/sea environment, this acts as a good stepping stone for testing viability of Deep VPR on the water.



Figure 5.2: Example Images from Symhpony Lake dataset. From Top Row to Bottom: Images taken from 2014, 2015, 2016 and 2017.

As the dataset was collected and stored as a set of sub-folders of image sets based on the day they were collected through 2014-2017, four folds are built for all image examples from each year, allowing us to use leave-one-out cross validation.

However, the initial size of this dataset is much larger than the Berlin image sets due to the higher capture frequency and the amount of runs, to bring this more inline I sub-sample images from each yearly fold to limit them to just 2500 images, this also introduces more variance between samples as each individual could be taken months apart despite being from the same location.



## 5.3.2 Architecture

### 5.3.2.1 SSM-VPR

SSM-VPR is a two-stage Deep VPR pipeline presented by Camara et al. [5]. The pipeline uses VGG16 as a backbone, with the Conv5\_2 and Conv4\_2 layers being used as output feature maps to be fed into stage 1 and 2 of the SSM-VPR pipeline respectively.

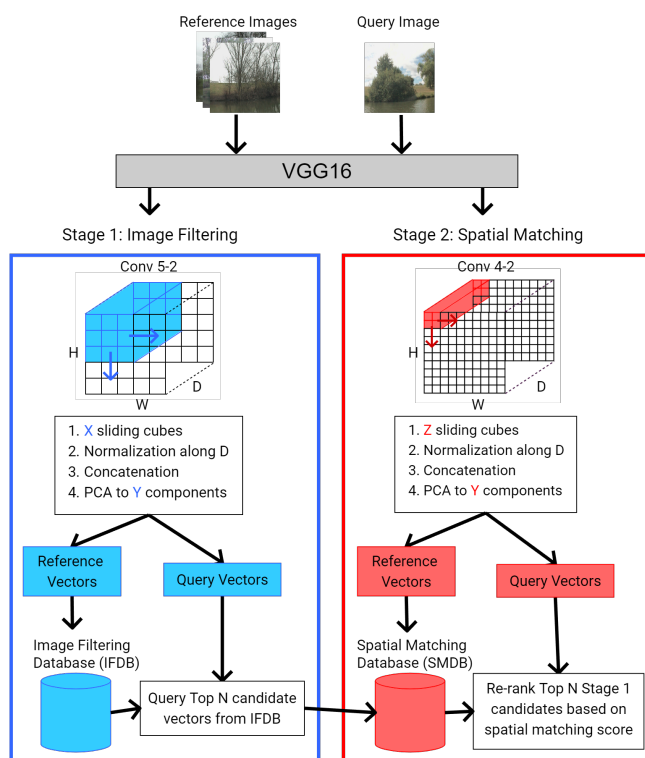


Figure 5.3: A diagram of the overall pipeline of SSM-VPR, including Stage 1: Image Filtering and Stage 2: Spatial Matching.

Before retrieval can take place, the search space must be built from an initial image set, the search space takes the form of two search databases for each stage. For stage 1, which focuses on semantic information extraction, Conv5\_2 feature maps based on input images to the CNN have a sliding window method applied with stride 1, for each region of the feature map that falls under this window, its values are normalized along the filter axis and concatenated into a large vector which is dimensionally reduced via PCA down to 100 channels as done so in the original SSM-VPR papers [5, 88].

This vector is then stored inside the Image Filtering Database (IFDB) alongside all other

vectors extracted from the feature map via the sliding window, with a complementary Image ID Database which tracks the initial input image ID that each vector was extracted from.

Stage 2 applies a similar sliding window based vector extraction method, the key differences being the larger resolution of the feature map produced by VGG16 Conv4\_2 and the much smaller size of the sliding window for this stage. This results in each vector recording much finer spatial information across the input image and these are stored in the Spatial Matching Database (SMDB), with its own Image ID database to keep track of image-vector relationships.

It should be noted that for both stages, PCA is initialized by extracting stage 1 and 2 vectors from random sub-sample of 250 images, which produce sets of feature vectors generated by stage 1 and 2 extraction that can be further sub-sampled to 2000 flattened vectors (For stage 1 and 2 vectors each) which are then used to initialize two individual PCA models for stage 1 and stage 2 dimensionality reduction.

Once the IFDB and SMDB are built, query images can then be passed through and have retrievals suggested via the whole two-stage pipeline. Stage 1 once again extracts several vectors from the query, then, in order to get a set of image candidates, a histogram with bins for each individual image within the search space is built. For each query vector, top  $N$  candidate vectors are retrieved from the IFDB via a euclidean distance based search and search space images associated with those candidates receive a point on the histogram, once all query vectors have been used for search the histogram is sorted into descending order and the top  $N$  images form an initial candidate list.

Given this list, stage 2 vectors are then extracted from the query image and the candidates vectors are acquired from the SMDB, for the query and each candidate their array of vectors are reshaped into a 2D vector array with the order being equivalent to the order in which the vectors were extracted via sliding window. A new histogram is made with each bin representing one of the stage 1 candidates.

Then a process known as spatial matching (See Figure 6.10) is performed, where for each candidate, anchor vectors are then identified between itself and the query, then, for each anchor pair, spatially equivalent vectors around the anchors are compared in order to measure spatial consistency. Every time these vectors are found to be a closest match, the candidate receives a point on the new histogram, after all checks are completed the histogram can once again be sorted into descending order, being used to re-rank the candidate list.

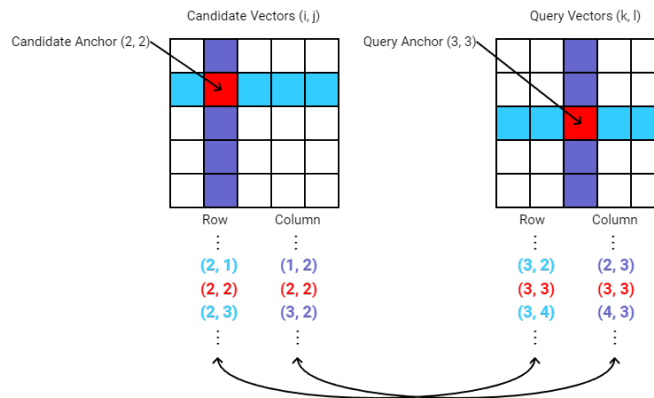


Figure 5.4: Figure inspired by the original paper [5]. A simplified representation of the spatial matching stage for a grid of query and retrieval vectors, taking a pair of anchor points between the two, their surrounding vectors along the row and column should also match if the features are spatially consistent.

### 5.3.2.2 Score-CAM

Score-CAM is a CAM-based saliency technique that focuses on identifying salient regions in an image through linearly weighted combinations of CNN activation maps. Score-CAM works by passing an input image through a CNN to generate a list of feature map filters, these filters are then upsampled and normalized to be used as masks for the input image, these masked images are then passed through the pipeline and a set of scores corresponding to each mask is generated.

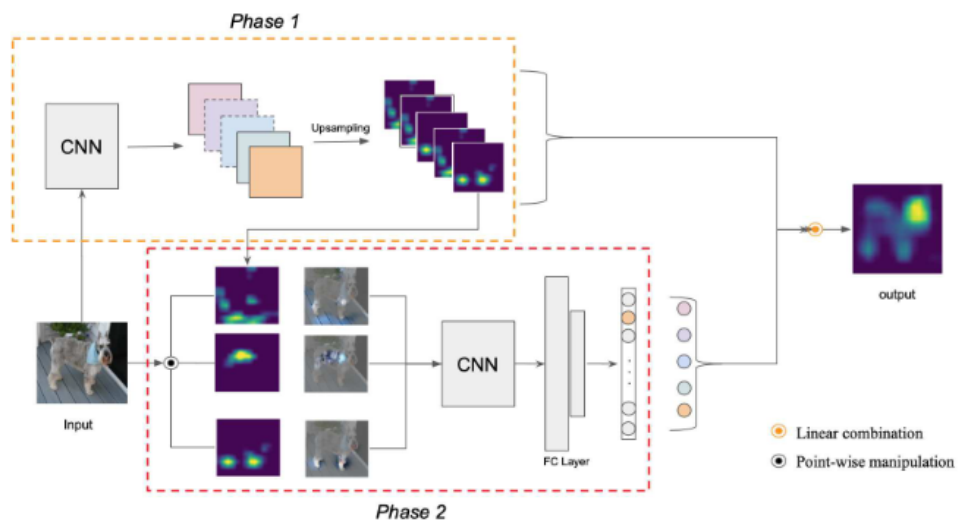


Figure 5.5: Score-CAM architecture from Wang et al. [26].

This method has several advantages over its predecessors in that it does not require any gradients and does not need to generate random masks, instead directly utilizing activation maps. Score-CAM has been shown to improve over previous methods such as RISE, Grad-CAM and Grad-CAM++ according to the deletion and insertion curve metric (i.e. The decrease and increase of prediction scores when weighted areas are removed/added from the images respectively).

While exploring implementations of Score-CAM, I encountered the `tf-keras-vis` code repository (<https://github.com/keisen/tf-keras-vis>), which implemented various saliency based methods in Tensorflow 2.0, including a modification of Score-CAM known as Faster Score-CAM.

Faster Score-CAM is an augmentation method for Score-CAM created by S. Tabayashi (<https://github.com/tabayashi0117/Score-CAM>), which limits the number of filters used for masking by ranking them in order of standard deviation and only taking the top  $k$  filters.

The method is influenced by Channel Pruning [177], whereby channels with constant value and thus low standard deviation can be identified and pruned from the list of activation maps used to generate the Score-CAM saliency map.

This benefits the final output as these more constant valued activation maps are not allowed to saturate the saliency values across the whole image, thus preventing specific features and structures from being highlighted.

Coincidentally, we found that this was an issue with the localization imagery which Faster Score-CAM was able to alleviate, thus we chose to make use of this novel approach.

For the SSM-VPR pipeline, I modified the formula so that the mean squared error between the array of SSM-VPR stage 1 vectors generated by the query and masked versions of the retrieval image are used as weights rather than class score, theoretically similar features between two images should minimize the distance between a query and positive image, so in this case any masks that increase an MSE loss between the two would indicate that an important feature was contained within the area covered by the mask.

### 5.4 Comparative Analysis and Results

I carry out a quantitative evaluation of the SSM-VPR model when applied to test folds from both datasets, each Berlin image set is treated as a simple train/test pair whereas the Symphony Lake image sets are divided into four leave-one-out cross validation folds where each test fold represents images from one year as described in my method.

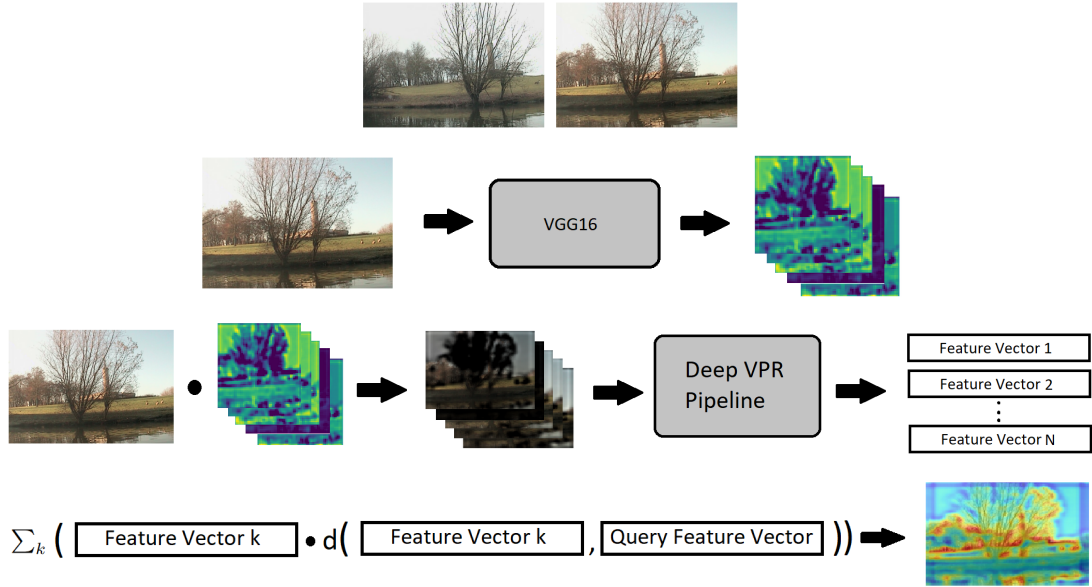


Figure 5.6: A simple diagram of Score-CAM adaptation for SSM-VPR. Top Row: A Query and Retrieval Image pair. Second Row: Retrieval Image is passed through CNN to get feature map filters. Third Row: The Retrieval is multiplied by these filters to produce masked versions, these are passed through the Deep VPR pipeline to obtain vectors for each masked image. Bottom Row: A distance metric is calculated between each masked vector and the queries vectors and used to weight the masks, the linear combination produces the saliency map.

This evaluation uses the precision-recall curve with an associated Area under curve (AUC) metric to measure overall performance on each test fold, as well as a Precision@TopK curve. A retrieval was considered a true recall if it fell within an acceptable radius of the query, the curve represents the change in these statistics as recorded retrievals are thresholded based on their prediction score, from maximum to minimum.

For now, I follow the same metrics for Precision and Recall as used in the SSM-VPR paper [5], where only the top retrieval is counted as the final output, precision is equal to  $\frac{TruePositive}{TruePositive+FalsePositive}$  whereas recall is equal to  $\frac{TruePositive}{TruePositive+FalseNegative}$ . In essence, for each score threshold I calculate the mean average precision metric for only the top retrieval results (mAP@1).

Finally, I also perform a visual observation of the query/retrieval outputs of each domain along with Score-CAM outputs in order to gauge general areas of saliency within the two image domains.

### 5.4.1 Quantitative Analysis

Here I present the quantitative results when applying SSM-VPR to my test folds from both land and waterborne domains in order to compare them against one another. The following figure 5.7 shows two statistics; PR-curves and Precision @ Top K.

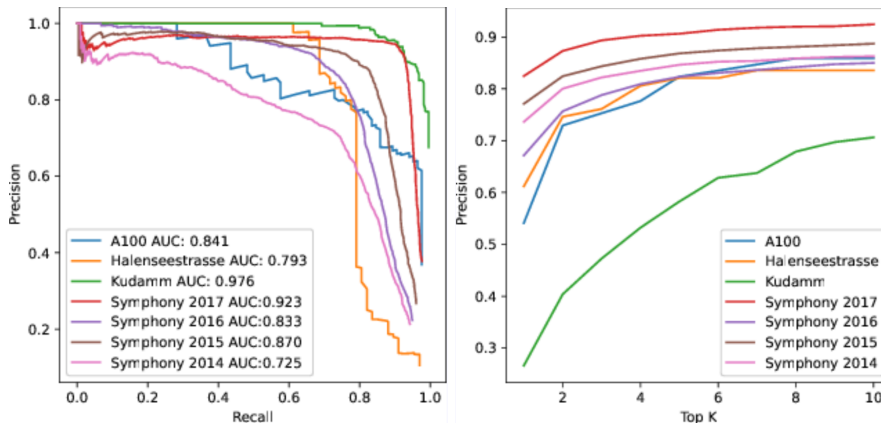


Figure 5.7: Left: PR-curves for Berlin and Symphony lake test folds. Right: Corresponding Precision at Top K curves.

Results in Figure 5.7 for the three Berlin image set test folds are consistently high as was expected, with each dataset maintaining a precision of 1 up to recall 0.8 for Kudamm, 0.6 for Halenseestrasse and 0.3 for A100, Kudamm is the most impressive with an AUC of 0.976 with A100 having the second highest AUC of 0.861 despite initially dropping off before Halenseestrasse, mainly due to it maintaining more of its precision after the initial drop off compared to Halenseestrasse which dips more sharply.

The results for each KFold of Symphony Lake are also high, having a more pronounced initial dip around precision 0.9 compared to the Berlin folds before becoming more stable and gradually progressing downwards before an expected sharper drop near 0.8 to 1.0 recall.

The more volatile curves of the Berlin datasets are likely a result of them having fewer query images to retrieve, meaning being able to retrieve individual images or small groups within the same area can cause sudden noticeable shifts in the curve, whereas the Symphony Lake datasets, even after sub-sampling, have much more query images in general as well as them being placed more densely. As such, these curves appear much smoother.

A major takeaway from these curves is that regardless of exact values and overall stability, each fold followed a similar pattern to one another, from which we can deduce that Symphony Lake images being waterborne did not have any unique impact on the Deep VPR models ability

to retrieve them.

The Precision at Top K curves are quite straightforward to interpret, increasing the number of candidates in stage 1 of SSM-VPR will always have a positive effect on all datasets up to  $K=10$ , after which the model experiences diminishing returns.

However, Berlin Kudamm has a noticeable dip compared to the other datasets for this statistic whereas in the PR curve it appears similar. This is likely because, as noted in previous works [5, 57], Kudamm is a particularly challenging dataset among the three Berlin sets due to distracting dynamic objects and the urban environment.

As for why the PR curve appears strong by comparison, remember that the PR curve analysis always made use of 10 Top K candidates before re-ranking hence the results will be more in line with the precisions seen on the far right of the Precision at Top K in Figure 5.7, and, at the highest score threshold, are able to surpass this precision albeit with lower recall.

This is because more stage 1 candidates allows SSM-VPR to re-rank them using the spatial matching methodology, without which we can see in the right plot of Figure 5.7 there is a clear minimum of precision at  $K=1$ , with each dataset fold having the largest jump in precision once a second candidate was introduced ( $K=2$ ) and thus re-ranking began to be performed.

### 5.4.2 Qualitative Analysis

Here I show examples of queries from the Berlin and Symphony Lake image sets alongside their retrievals. After this I will show what features/areas of the top retrievals are highlighted as being salient by Score-CAM.

In figure 5.8 all three queries have a top-scoring ground truth retrieval followed by less accurate retrievals as the retrieval rank decreases down from 1st to 3rd place, which is to be expected from a Deep VPR pipeline, with that in mind, let us see what features in the top retrievals were considered salient using Score-CAM:

When looking at the saliency, Score-CAM tends to weight areas containing similar objects between the query and retrieved positive quite well, although for the result in row 2 of figure 5.9 there is far too much focus on the biker, which would indicate that some of the activation maps are vulnerable to being distracted by obstructive objects.

In order to get these mappings I had to reduce the number of activation maps used for the masks using the Faster Score-CAM to 10, as using too many would cause the saliency map to appear too broadly saturated, implying that for localized imagery only a small handful of high deviation filter maps are useful for human interpretability.

## 5. Evaluating Waterborne Deep VPR

---



Figure 5.8: 3x4 Table depicting a series of queries and associated image retrievals row-by-row from the Berlin Halensee strasse image set. First Column: Image Queries. Second-Fourth Column: 1st-3rd highest scoring retrievals for each query.

Moving on to Symphony Lake, Figure 5.10 shows that although landmarks are generally more sparse in this domain, for each query the rank 1 through 3 retrieval images are all ground truth if not near, although this may be helped by the higher density of Symphony Lake samples even after downsampling the dataset.

Finally, Figure 5.11 shows the Score-CAM saliency results when applied to the 1st place retrievals from Figure 5.10, results for this are promising as the saliency consistently highlights notable features along the top of the visible landline within each image, including buildings and recognizable tree lines. These results are quite promising and could greatly aid in making SSM-VPR's decision making process more interpretable to end users.

### 5.5 Summary

In this work, I demonstrate the performance of a modern deep visual place recognition architecture on a waterborne dataset, and, compared it to the models performance on traditional land-based datasets to gauge it's viability to perform waterborne place recognition.

Through quantitative analysis, I conclude that given ample data, deep VPR architectures can perform just as well on small-scale waterborne datasets as their land-based counterparts, with concerns regarding the lack of man-made structures and over representation of vegetation



## 5. Evaluating Waterborne Deep VPR

---



Figure 5.9: 3x3 Table depicting Query Images (First Column), their rank 1 retrievals (Second Column) and the result of applying Score-CAM to the rank 1 retrievals (Third Column).

largely being addressed, as SSM-VPR is perfectly capable of matching images containing just simple banks and vegetation across different angles, seasons and lighting conditions.

I also aimed to show the potential of explainable AI when applied to both domains, Score-CAM performs quite well despite the task-based shift to place recognition as opposed to image classification explainability, with the Faster Score-CAM filter map selection method being key in acquiring interpretable saliency maps. However, reducing the number of filters used for masking by such a large degree means that some information may be getting lost, so a refinement of the algorithm for calculating place recognition saliency may be necessary.

Although Symphony Lake is a good starting point for waterborne imagery due to its overall image capture density along the area as well as the frequency of runs over four years, it is still technically more of a bucolic image set compared to the larger shorelines I wish to explore. This means it does not contain certain challenging features that one would encounter in a more long-range shoreline image set such as camera water obstruction, movement of the boat influencing the camera angle and variable atmospheric visibility.

As such, follow-up work would focus on building an open-source shoreline image dataset to further test deep VPR's potential in the waterborne domain and look into improved ways of integrating saliency map techniques into SSM-VPR's architecture to improve explain ability.

## 5. Evaluating Waterborne Deep VPR

---



Figure 5.10: 3x4 Table depicting a series of queries and associated image retrievals row-by-row from the Symphony Lake image set. First Column: Image Queries. Second-Fourth Column: 1st-3rd highest scoring retrievals for each query.

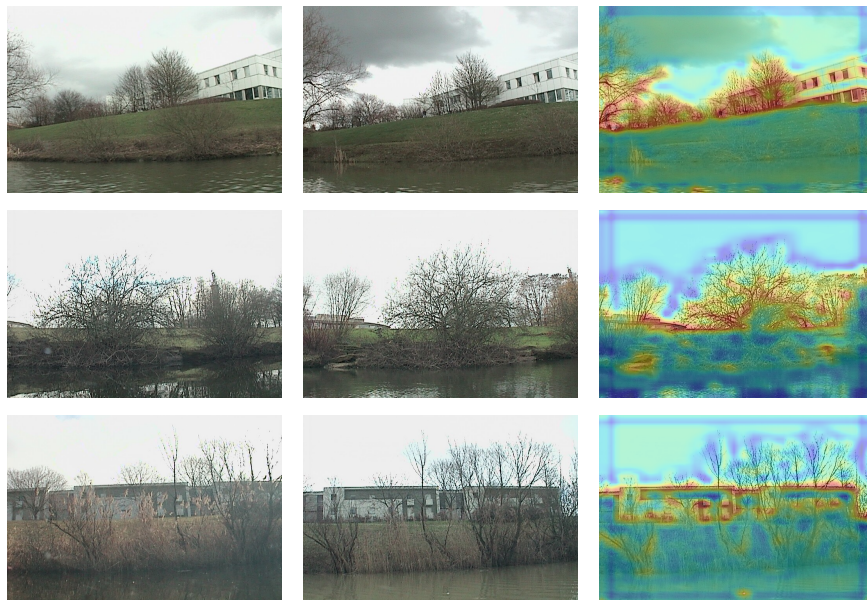


Figure 5.11: 3x3 Table depicting Query Images (First Column), their rank 1 retrievals (Second Column) and the result of applying Score-CAM to the rank 1 retrievals (Third Column).

## Chapter 6

# Improvement of Waterborne Deep VPR

### Contents

---

6.1	Introduction . . . . .	<b>104</b>
6.2	Proposed Approach . . . . .	<b>105</b>
6.3	Datasets . . . . .	<b>106</b>
6.3.1	Plymouth Sound Dataset . . . . .	106
6.3.2	Symphony Lake Dataset . . . . .	106
6.4	Architectures . . . . .	<b>106</b>
6.4.1	SSM-VPR (Baseline) . . . . .	106
6.4.2	SSM-VPR w/ Stage 1 Selective Search Region Proposal . . . . .	107
6.4.3	SSM-VPR w/ Stage 1 rOSD Region Proposal . . . . .	108
6.4.4	SSM-VPR w/ Stage 2 WaSR Semantic Line-based Region Proposal (SHM-VPR) . . . . .	108
6.5	Comparative Analysis and Results . . . . .	<b>111</b>
6.5.1	Quantitative Analysis . . . . .	111
6.5.2	Qualitative Analysis . . . . .	114
6.6	Summary . . . . .	<b>120</b>

---

## 6.1 Introduction

Having acquired an in-house waterborne image set (See Chapter 4) after my work in Chapter 5, I propose a novel Deep Visual Place Recognition pipeline that minimizes redundant feature extraction and maximizes salient feature extraction.

My approach is largely informed by the unique nature of waterborne imagery, namely the tendency for salient land features to make up a minority of the overall image, with the rest being disposable sea and sky regions. I initially attempt to exploit this via unsupervised region proposal, but I later propose a horizon-based approach that provides improved performance.

To that end, I present quantitative results from a set of experiments with the Semantic and Spatial Matching Visual Place Recognition (SSM-VPR) pipeline [5] that attempt to maximize performance on the Plymouth Sound dataset. I modify SSM-VPR into a set of separate versions and apply each to test folds based on Plymouth Sound as well as the Symphony Lake dataset [58] to facilitate a comparison between shoreline and bucolic waterborne imagery.

I modify SSM-VPR in a number of ways, finding that the most effective modification of the pipeline for dealing with shoreline imagery is to encourage structural consistency of features along the visible horizon between a query and retrieval, creating a novel pipeline dubbed Semantic and Horizon Based Matching Visual Place Recognition (SHM-VPR).

Before SHM-VPR, I theorized that unsupervised region proposal could also aid in limiting the pipelines feature perception to important shoreline features, perhaps being able to mimic real world navigation techniques where landmark identification is preferred over a more computer-like brute force search. I experiment with two separate unsupervised region proposal methods, Selective Search [13] and a saliency-based method proposed by Vo et al. for their rOSD paper [127].

My SHM-VPR method uses a similar procedure as the WASABI network [176] in that it extracts a semantic line from visible land, the locations of which are predicted using semantic segmentation. As the only major semantic class of interest for us is “land” and the image domain is waterborne, I make use of the WaSR segmenter [37] which is purpose built for such a domain and classifies pixels as land, sea and sky.

WaSR also allows us to predict which images are largely devoid of land information, this is useful as images that fall into this category (i.e. Open Sea) are not suitable for place recognition as they lack enough information to be reasonably retrievable. The main issue this creates is that when provided to Deep VPR as a query image these examples will artificially reduce the pipelines metric scores, as they will most likely result in a set of false positive retrievals.

This makes proper quantitative analysis difficult as it offsets the metrics, manually labeling all images containing little to no land information could be an option, but this would require significant time investment and may not be scalable, WaSR however allows us to get a predicted percentage of land pixels within each image automatically and thus helps to solve this issue, allowing us to filter allowed query images by thresholding them based on land pixel percentage.

Ultimately, I find the SHM-VPR pipeline to provide state-of-the-art results on the Plymouth Sound image dataset, although it is a domain-specific pipeline, and does not translate to inland locational imagery.

## 6.2 Proposed Approach

As the original SSM-VPR did not rely on any training or fine-tuning, simply using pre-trained VGG16 on Places-365 as backbone, I continue using this backbone as is rather than retraining, instead focusing on making structural changes.

To add on to this, the Plymouth Sound dataset is not labeled in the same way that Places-365 is, with the latter being labeled by “scene” for scene classification training [178] (i.e., bar, shop, stairway, etc.), as such it does not facilitate the same classification training originally used for the backbone. Creating such labels manually for all 16673 images would also be a significant time investment.

Moving on to the proposal, first is a quantitative evaluation of four versions of the SSM-VPR pipeline on both the Plymouth Sound dataset as well as Symphony Lake for the sake of comparison. These four versions include the original SSM-VPR (i.e. Baseline), SSM-VPR with Selective Search for stage 1 region-based vector selection, SSM-VPR with Vo et al.’s method for stage 1 region-based vector selection and SHM-VPR, which uses a semantic line generated from WaSR land pixel prediction to guide stage 2 region-based vector selection.

I will then show the necessity of automated land, sea and sky based pixel predictions for conducting stable Waterborne Deep VPR evaluation, especially within more long-range shore-line areas such as Plymouth Sound which have access to open sea.

Finally, I show a more in depth qualitative comparison between the different techniques employed for both stage 1 and 2 of SSM-VPR, going over the pros and cons of each method and how it interacts with the features generated by the CNN backbone before storing them as vectors.

## **6.3 Datasets**

### **6.3.1 Plymouth Sound Dataset**

Described in-depth in Chapter 4, the Plymouth Sound dataset consists of several traversals along the ares of its namesake captured by the IBM/Promare Mayflower Autonomous Ship (MAS). Traversals begin at Turnchapel Wharf and cover differing areas of Plymouth Sound including Drake Island, Cawsand Bay, Rame and Whitsand Bay.

There are seven original runs in total taken between 30/03/2022 and 14/04/2022, so for evaluation I divide the overall image set into seven leave-one-out cross validation test folds. For some folds that stray especially far from Plymouth Sound such that their training folds do not have any matching ground truths for some queries, I only evaluate those queries that do have possible ground truths within said training folds according to some ground truth radius.

As previously discussed, the dataset is quite challenging due to large variations in the distance of the camera to the visible shore as the MAS travels outwards into the Plymouth Sound, it also has more general viewpoint/feature changes for most locations based on the day recorded and is very sparse in terms of land features for performing place recognition. Average land pixel percentage across all images is around 5% according to WaSR, influenced heavily by the amount of images capturing open sea.

### **6.3.2 Symphony Lake Dataset**

The same dataset used in Chapter 5.3.1, consists of weekly runs along Symphony Lake in Metz, France, by a small autonomous vehicle from 2014-2017. In order to match the same number of folds as the Plymouth Sound dataset for this Chapter I instead select seven random folders from the overall dataset representing one run each between 2014-2017 and use the same leave-one-out cross validation setup to get seven test folds.

This should ensure that both datasets have a similar level of density when it comes to the position of image captures, relative to the overall distance each dataset covers.

## **6.4 Architectures**

### **6.4.1 SSM-VPR (Baseline)**

Original SSM-VPR pipeline outlined in Camara et al. [5], the only domain-specific change I made was that the lack of meaningful features along the edges of the images resulted in an

increase in the activation values along the edges of the feature map due to the use of same padding in the convolutional layers, to reduce the impact of these sections the sliding window will not cover the minimum and maximum edge of the feature map.

### 6.4.2 SSM-VPR w/ Stage 1 Selective Search Region Proposal

A modification to baseline SSM-VPR whereby instead of using sliding window for vector extraction in stage 1, I instead apply the Selective Search algorithm to the feature map to get a set of region proposals that are then pooled and used to build IFDB vectors for SSM-VPR stage 1 search.

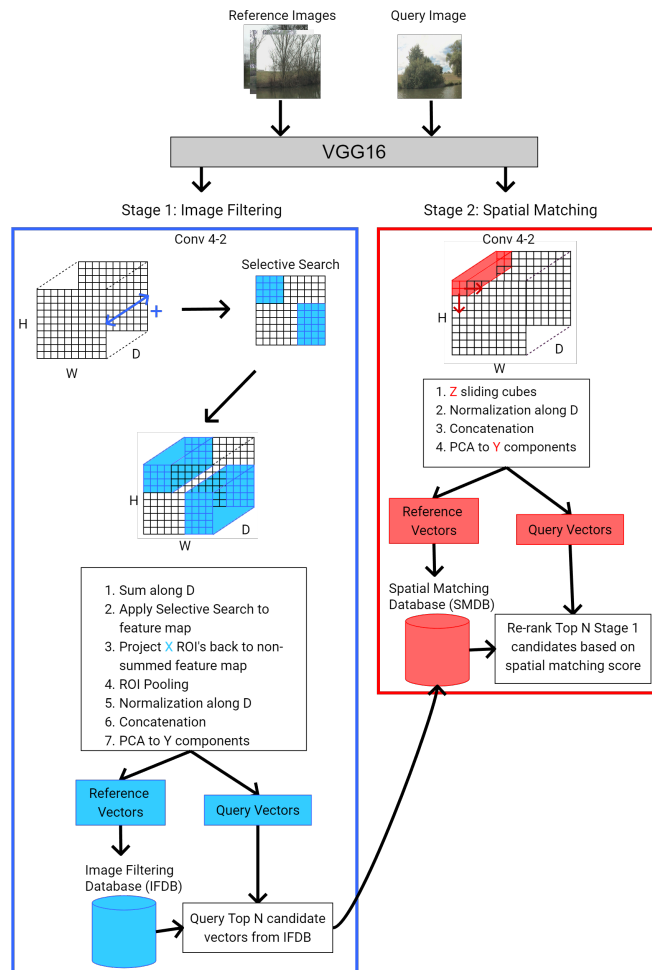


Figure 6.1: Diagram of the proposed SSM-VPR w/ Stage 1 Selective Search Region Proposal, which takes a set of region proposals from selective search based on a 2D pseudo-image and uses them to select sub-regions of the feature map for vectorization.



Selective Search is applied directly to the feature map to save computational time, this is done by aggregating the feature map along its filter axis into a 2D pseudo-image, the top  $N$  regions are passed through an ROI pooling layer much like in Fast-RCNN [15] to maintain spatial consistency for vectorization. The dimensions of the pooled feature map regions are equal to those extracted via the sliding window approach.

Early results indicated that pseudo-images generated from feature maps of lower resolution stage 1 image Conv5\_2 outputs were not sufficient for Selective Search, so instead I use the higher resolution stage 2 image Conv4\_2 outputs for both stages in order to ensure effective region proposal-based vectors are used for stage 1.

### **6.4.3 SSM-VPR w/ Stage 1 rOSD Region Proposal**

Fundamentally identical to the Selective Search modification, only now I use the rOSD region proposal method instead, which was built to work on 2D pseudo-images generated from convolutional feature maps originally and as such translates to the task more naturally.

In Vo et al. [127] two recommendations are made, that the final list of region proposals should be a combination of proposals generated from at least two convolutional layer outputs to capture objects of various scales and that the most suitable layers for VGG16 are Conv5\_3 and Conv4\_3.

As such, I change the output convolutional layers of stage 1 and 2 of SSM-VPR to Conv5\_3 and Conv4\_3 respectively, when calculating region proposals for stage 1 I build 2D pseudo-images from the output feature maps of both stages and apply the rOSD region proposal method to each and take  $\frac{N}{2}$  proposals to get  $N$  in total.

Proposed regions are all applied to the stage 1 SSM-VPR feature map and pooled to the same dimensions as before for vectorization.

### **6.4.4 SSM-VPR w/ Stage 2 WaSR Semantic Line-based Region Proposal (SHM-VPR)**

This model keeps stage 1 of SSM-VPR the same and focuses on making edits to stage 2. This method is dependent on the WaSR segmenter and leverages its prediction mask to extract a set of coordinates that represent a semantic line along the visible land in the image.

Using these coordinates, stage 2 applies a sliding window that only moves across the  $x$  axis of the feature map once, with the  $y$  coordinate at each step being determined by projecting the horizon line onto the feature map and getting a set of approximate coordinates.



## 6. Improvement of Waterborne Deep VPR

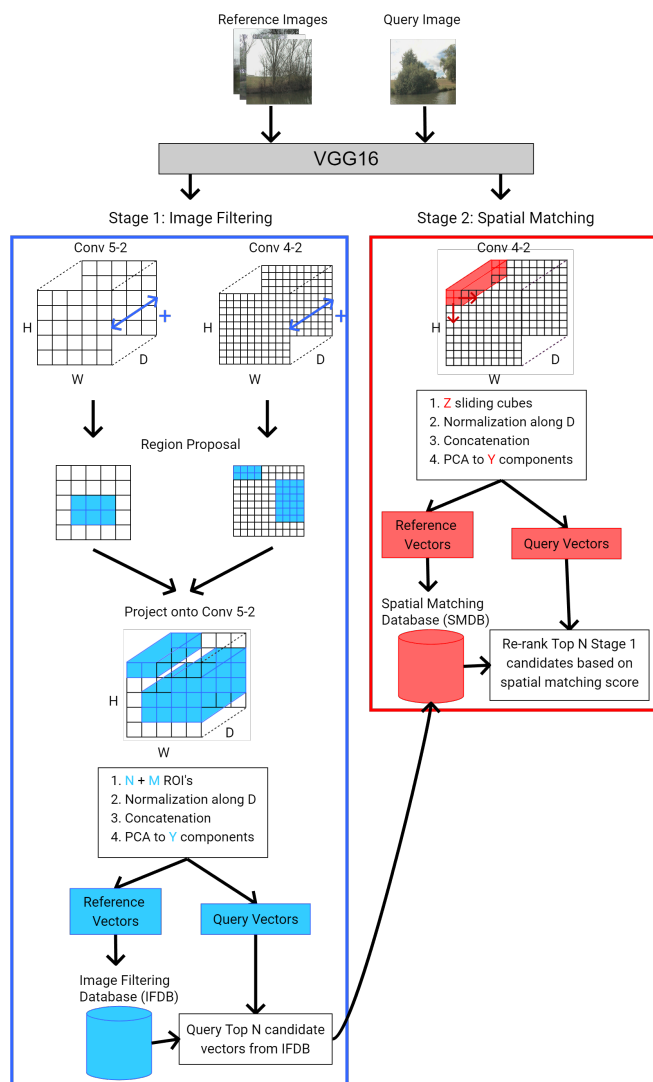


Figure 6.2: Diagram of the proposed SSM-VPR w/ Stage 1 rOSD Region Proposal, which takes a set of region proposals from the rOSD method developed by Vo et al. based on 2D pseudo-images from both stages and uses them to select sub-regions of the feature map for vectorization.

This leaves us with a single row of stage 2 vectors as opposed to a grid, exponentially reducing the overall number of vectors in the SMDB. The spatial matching stage, which checks for closest neighbour consistency around spatially arranged anchor vectors between a query and retrieval, still works as before but now only needs to make spatial matches along one dimension rather than two.

Development of this method was motivated by close examinations of CNN feature maps extracted from the Plymouth Sound dataset over the course of the project, where it was found

## 6. Improvement of Waterborne Deep VPR

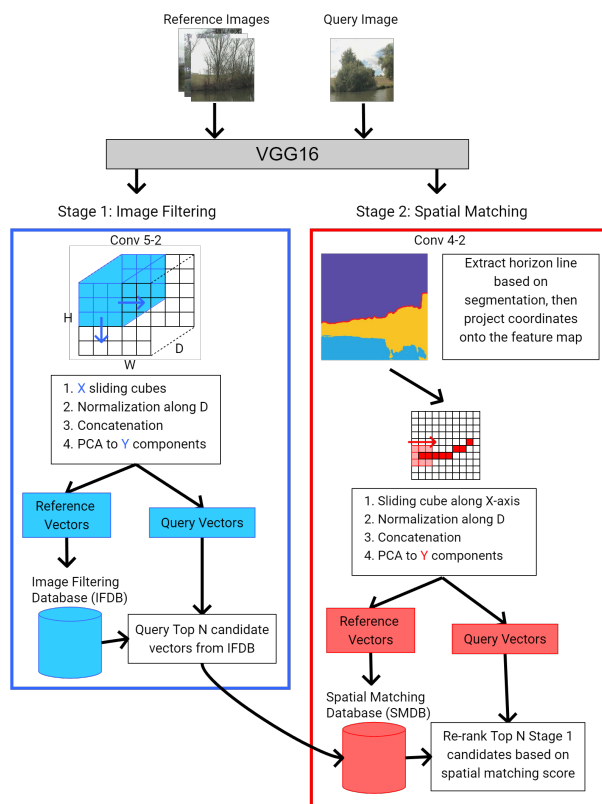


Figure 6.3: The SHM-VPR pipeline, here SSM-VPR stage 1 is kept the same as baseline but stage 2 now uses an estimated horizon line based on WaSR and projects it onto the feature map, the sliding window then moves along the map in a single row across the x-axis, using the y coordinate of the projected horizon line at each step.

that the most consistently activated features were those that fell along the semantic line associated with visible land, similar to the findings made by Benbihi et al. [176] for bucolic environments.

For land-based domains this would not be totally beneficial as various landmarks and building features tend to appear below the top of the visible land line, which in these domains would typically be rooftops. However, for long-range waterborne images like those seen in Plymouth Sound, many buildings and other features below the land line take up such a small portion of the image that when the image is reduced to a manageable size for a Deep CNN (i.e. 224x224, 448x448), they hold little presence within the final feature map.

Taking the waterborne domain into account, this model makes it so that the re-ranking stage is more focused on spatially matching only the most visually apparent and variable structure of these images, especially within long-range shoreline areas.

## 6.5 Comparative Analysis and Results

### 6.5.1 Quantitative Analysis

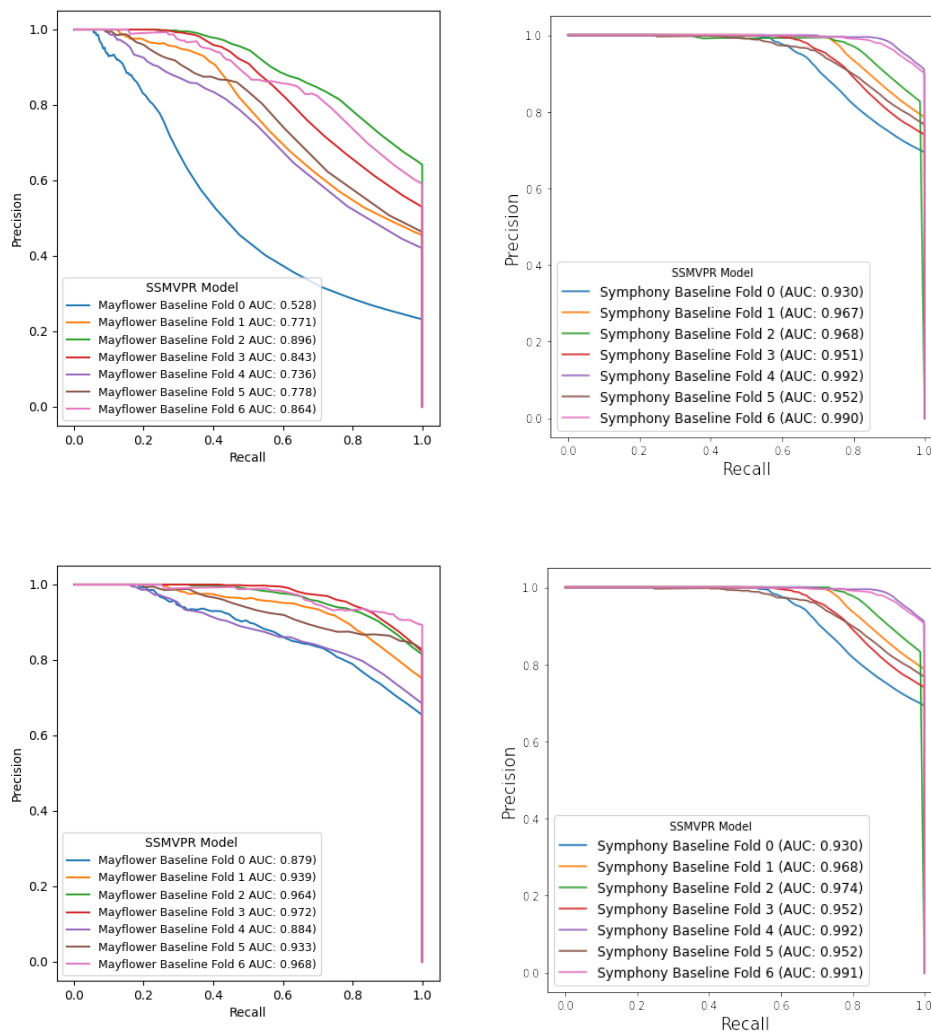


Figure 6.4: Top Row: Initial PR Curve and AUC metrics for seven Plymouth Sound (Left) and Symphony Lake (Right) using Baseline SSM-VPR. Bottom Row: The same metrics after thresholding query images based on WaSR predicted land pixel percentage.

After evaluating all seven test folds of both datasets with baseline SSM-VPR using the precision-recall curve with respect to mAP@1 and the AUC metric, the top row of Figure 6.4 shows that results are consistently high for Symphony Lake, which is consistent with my

## 6. Improvement of Waterborne Deep VPR

---

findings in Chapter 5, however, for Plymouth Sound there is a great deal of variation in the metrics for each fold which at first appears quite troubling.

The reason for this variance is something that I alluded to earlier in this chapter, that being the prevalence of many empty, open sea images within the Plymouth Sound image set. Each test fold represents a different traversal and between these the amount of these “empty” images differs. Because each fold has a different number of these and because said images cannot typically be retrieved due to a lack of features, the metrics for each test fold are effectively capped based on the percentage of actually retrievable images within the fold.

Thankfully, this can be alleviated by using pixel-wise label predictions from WaSR, from which I can get the overall percentage of land pixels, and therefore useful features, within an image. Through prior testing I found the average land percentage for all images was around 5%, so I use this as a threshold for which queries are taken into consideration for the evaluation metrics, which can be seen in the bottom row of Figure 6.4.

The effects of thresholding on Mayflower images are as follows:

Fold	Images before threshold	Images After threshold
0	2760	637
1	2268	1091
2	1380	1017
3	2382	1307
4	4224	1084
5	3066	781
6	593	335

Whereas the effects on Symphony Lake can be seen here:

## 6. Improvement of Waterborne Deep VPR

---

Fold	Images before threshold	Images After threshold
0	2723	2714
1	2487	2480
2	916	906
3	1667	1666
4	1874	1871
5	2208	2203
6	2039	2019

As you can see, the thresholding is negligible on Symphony Lake data but has massive impacts on the Mayflower imagery, sometimes reducing the amount of eligible images by two thirds, highlighting the sheer amount of water and sky only images contained within this set.

With this threshold applied, I get a more helpful and interpretable set of performance metrics across the Plymouth Sound test folds, which I can now see that, when evaluating valid queries, performance is comparable to Symphony Lake.

With that issue settled, I move on to comparing performance on both datasets using my four pipeline versions. I apply the same thresholding to each test fold and, for the sake of keeping the performance metrics succinct, I calculated the PR curve and AUC values based on all seven test folds for each dataset at once to get an average for each individual pipeline. Results are shown in Figure 6.5.

Figure 6.5 shows that, unfortunately, the Selective Search and rOSD region versions for stage 1 region proposal have lesser performance than Baseline for both datasets according to the AUC values. For reasons I will discuss in the qualitative analysis, it is clear that although still functional, the ability of these pipelines to propose meaningful regions of varying positions and sizes as opposed to the fixed sliding window regions of Baseline falls short.

These methods also add additional complexity to the pipeline and thus inference time is slower on average, so without any further developments I deem these unsupervised region proposal methods unsuitable for Waterborne Deep VPR pipeline integration.

My fourth pipeline, SHM-VPR, manages to surpass Baseline performance on Plymouth Sound but has worse performance on Symphony Lake. So, this method does present a slight improvement for this works ideal waterborne image domain of long-range shoreline, but does not necessarily translate to the short-range bucolic imagery of Symphony Lake.

## 6. Improvement of Waterborne Deep VPR

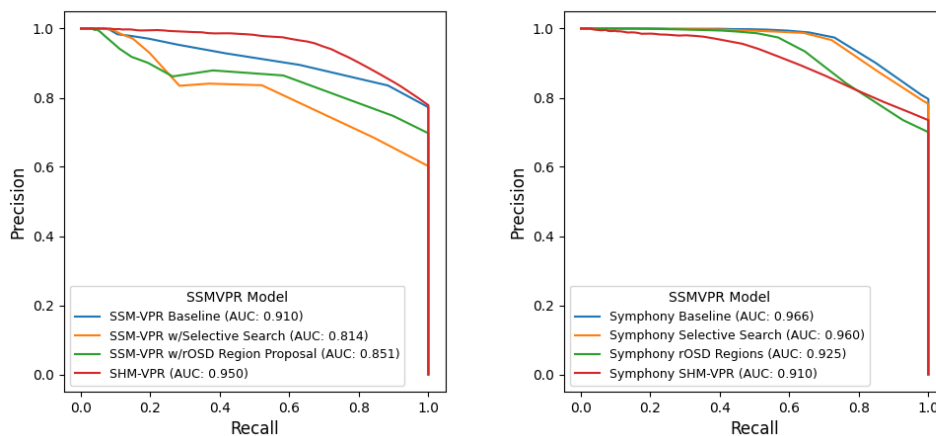


Figure 6.5: Overall PR Curves for my four pipeline versions on Plymouth Sound (Left) and Symphony Lake (Right). Baseline model and Unsupervised Region Proposal variants are consistent across both sets, SHM-VPR achieves greater performance on Plymouth Sound but lesser performance on Symphony Lake.

However, SHM-VPR presents additional advantages beyond just raw performance, in the appendix of Camara et al.'s original paper it was noted that the storage space required for SSM-VPR's SMDB was considerable, and I myself found this to be the case with the Plymouth Sound dataset. The main driver of this is the fact that the SMDB vectors are extremely numerous per image, SSM-VPR stage 2 sliding window approach extracts a total of  $54 \times 54 = 4608$  vectors, based on a  $56 \times 56$  VGG16 Conv4\_2 output feature map calculated from the recommended  $448 \times 448$  input image resolution.

SHM-VPR reduces the amount of vectors needed for stage 2 exponentially and thus results in an equivalent reduction of storage space needed for said vectors, because SHM-VPR stage 2 only extracts a single 54 length row of vectors. It also reduces the dimensional complexity of the stage 2 spatial matching algorithm.

### 6.5.2 Qualitative Analysis

Having gone over the performance metrics of each pipeline, I carry out a qualitative analysis of the retrieval results by looking at visual representations of the inner workings each pipeline, specifically when applied to the Plymouth Sound imagery as this is my main image domain of interest.

Figure 6.6 shows a set of true positives, viable query images which the pipeline retrieved

successfully, false positives, viable query images that were not retrieved successfully and true negatives, unviable query images as determined by WaSR land pixel percentage threshold them based on.

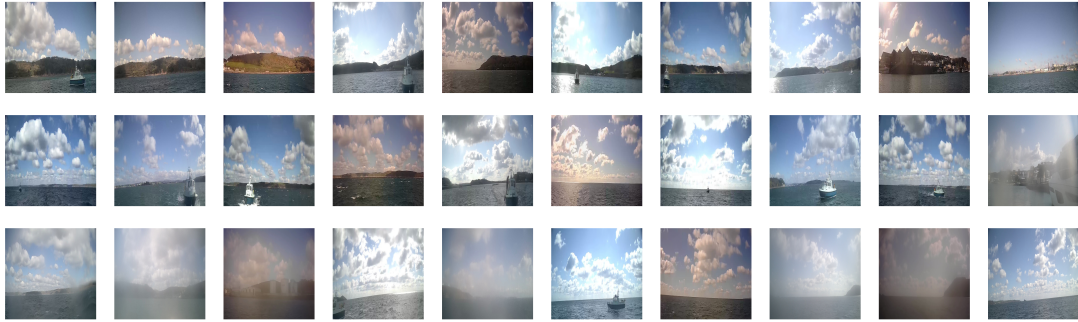


Figure 6.6: Example queries from the Plymouth Sound image set, divided into true positives, false positives and true negatives. Top Row: True Positive Queries. Middle Row: False Positives Queries. Bottom Row: True Negative Queries.

Looking at the true positive set, most images in this category are, unsurprisingly, those with clear shots of local shoreline, with recognizable shapes and minimal obstructions. The false positive set contains some images erroneously labeled as valid due to interference from objects such as boats, which WaSR identifies as land, boosting their land pixel percentage; this set also contains some blurred/obstructed images as well as more clear cut fail cases.

The true negatives give a good example of feature devoid images, most of these are either pointed directly out at sea where there is no land information, are severely lacking in shoreline features or are so blurred due to water obstruction that WaSR does not identify the present features as land.

This gives us a general view into what images the pipelines are able to retrieve successfully/unsuccessfully as well as a look into how lack of features, it also shows how water obstructions can make waterborne imagery unviable for image retrieval.

Beyond this, the following subsections will show how the pipelines interact with input images and image feature maps for the sake of comparison, providing a greater insight into the performance results.

### 6.5.2.1 Comparison of Different Approaches to SSM-VPR Stage 1 Region Selection

Starting with Baseline SSM-VPR, Figure 6.7 depicts how, given a query image, stage 1 of SSM-VPR takes the activation map and divides it into a set of fixed regions via sliding cube.

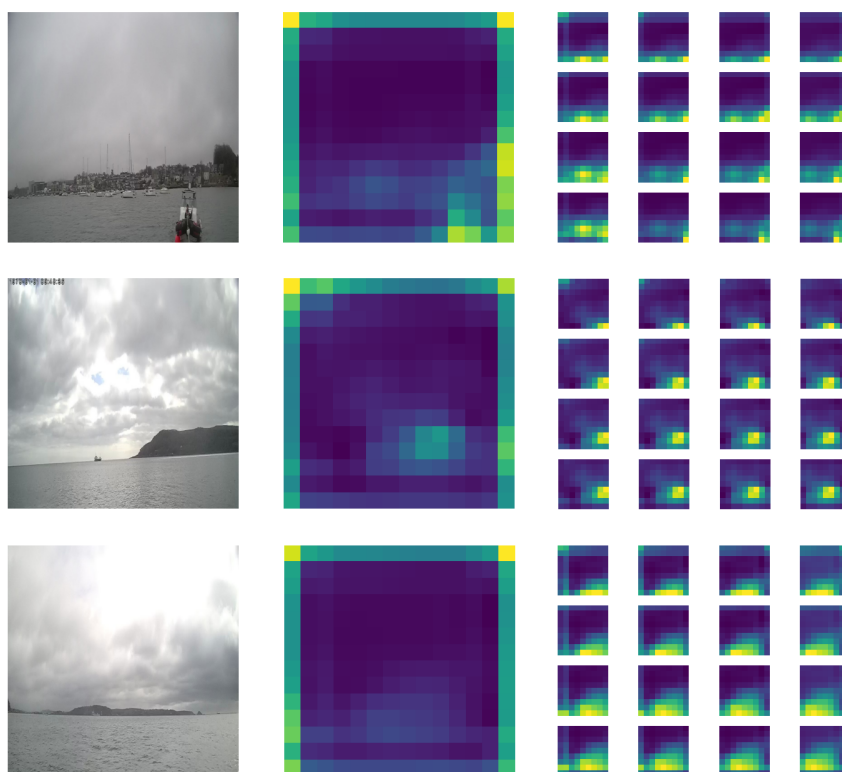


Figure 6.7: Example of Baseline SSM-VPR semantic regions: For each input image, the VGG16 activation map is divided into a set of sub-regions via sliding window. I show these feature maps in 2D by summing along the filter axis.

The visualisation shows that each region acts as a slight perspective shift of the overall image.

When IFDB vectors are matched to each individual vectorized region and rank their associated retrieval images via histogram score, SSM-VPR ensures that each image retrieval must match the query across multiple perspectives, incentivizing retrieval images that not only contains the same features as the query, but also views them from a similar perspective and thus from a similar position.

For Selective Search based SSM-VPR, we receive a number of suggested sub-regions which are then used for SSM-VPR stage 1 region based vector extraction. This means that instead of each sub-region representing a slight perspective shift of the overall image, each one now represents an area of interest which in theory should be similar to how mariners point out a series of landmarks.

From the previous section one can see that this method does not perform as well as the baseline, the reason for this can be seen in Figure 6.8, where the selective search algorithm's



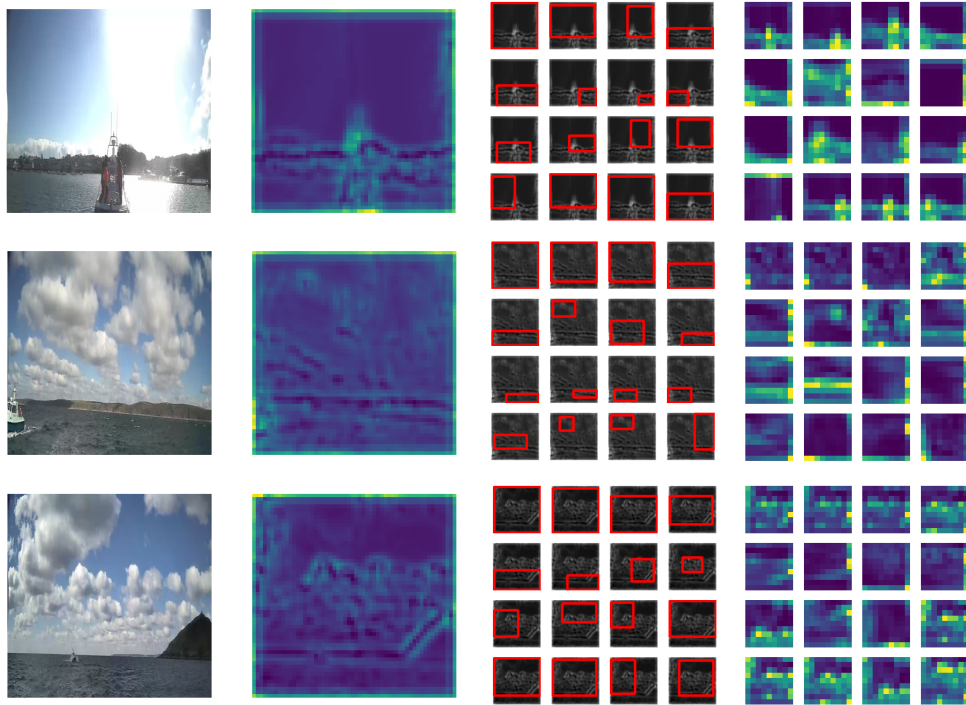


Figure 6.8: Example of SSM-VPR with semantic regions based on Selective Search: For this image, the activation is summed along the filter axis, then selective search region suggestions are made based on this to inform the extraction of feature map sub-regions.

attention is often drawn away from the land strip by areas of sea and sky.

There is also object interference, in the example on the first row a boat appears in the image and remains visible in the activation map. Objects like this draw attention from the selective search algorithm, which is undesirable for place recognition as it is a variant feature, the boat could simply move or not be visible at any time if a picture of the location were to be taken again, adding erroneous data to the vectors.

Overall it appears selective search is not able to discern the types of sub-regions across the shoreline, likely due to it being unsupervised and therefore not geared specifically to the shoreline image domain.

As I have discussed, I also tested the rOSD region proposal method integrated into the SSM-VPR pipeline as the third modification, which was done largely to verify the effectiveness of unsupervised region proposal in general, however I now know that this also failed to compete with baseline SSM-VPR.

Given an example of rOSD Region Proposal suggestions on a feature map, Figure 6.9

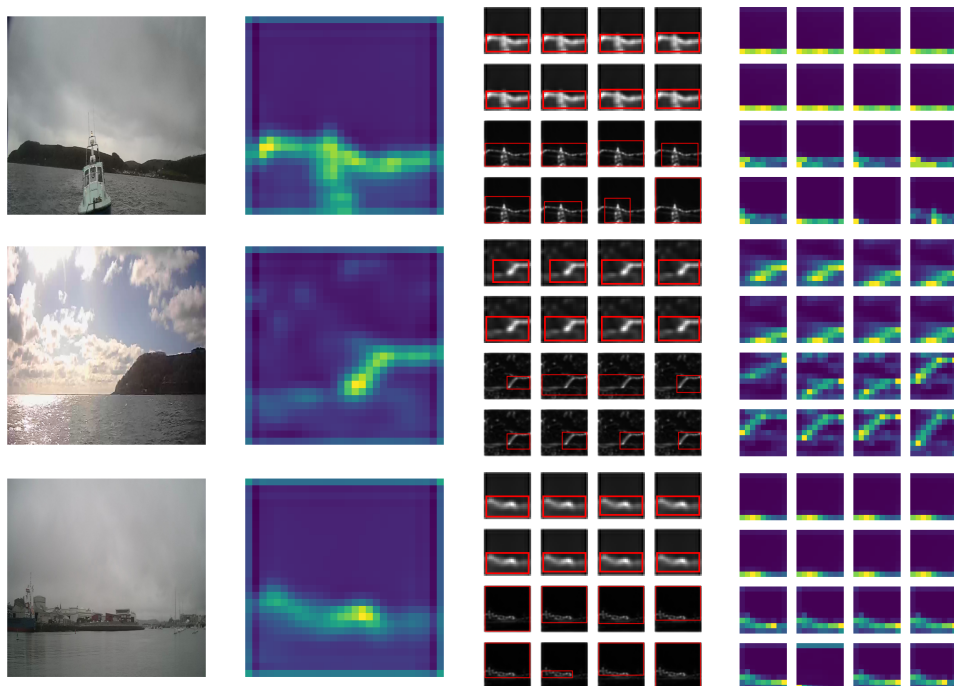


Figure 6.9: Example of SSM-VPR with semantic regions based on the rOSD paper region proposal method: This figure is identical in terms of presentation to Figure 6.8. Interesting to note is that in the third column it can be seen that the rOSD algorithm always takes an even number of region proposals from both the Conv5\_3 and Conv4\_3 generated 2D pseudo images.

shows that the method does single out the shoreline quite effectively, however the issue here is one of redundancy, most of the regions are simply repeats of each other. This means the number of unique sub-regions and thus perspectives of the shoreline is lower even though the most notable area has been covered.

These examples also showcases that ROI pooling these regions to maintain consistency may not be ideal - many of the patches have seemingly lost their distinctive shapes once pooled to 9x9 and are reduced down to simple edges.

### 6.5.2.2 Comparison of Different Approaches to SSM-VPR Stage 2 Region Selection

A major part of SSM-VPR stage 2 is the formation of a larger grid of much finer vectors for each image compared to stage 1, for each retrieval these are used for the re-ranking stage depicted in Figure 6.10, which scores them based on the number of spatial consistency matches across a set of anchor vectors.

This method is very effective for re-ranking as it ensures the top retrieval is highly spa-

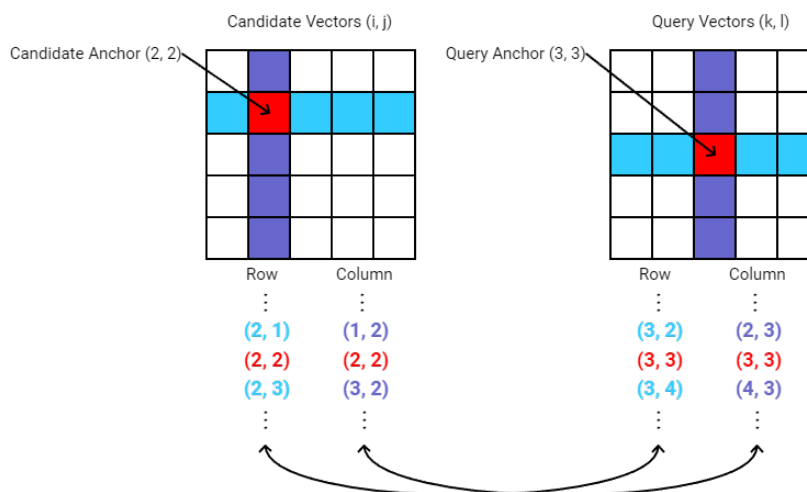


Figure 6.10: Figure inspired by the original paper [5]. A simplified representation of the spatial matching stage for a grid of query and retrieval vectors, taking a pair of anchor points between the two, their surrounding vectors along the row and column should also match if the features are spatially consistent

tially equivalent with the query, making it more likely that the image was captured from a similar location/perspective. However, I hypothesized that because many long-range shoreline images contained empty space and the sub-regions these vectors represent are not as broad, the likelihood that a great deal of the grid was made up of vectors representing empty space was high.

These redundant vectors hamper the model in two key ways, they inflate the storage requirements of the SMDB and their inclusion in the spatial matching calculation reduces inference speed. The redundant vectors could also be negatively impacting the PCA initialization that SSM-VPR relies upon for dimensionality reduction as the initialization batch is based on extracted vectors from a random sample of reference images and therefore could be influenced by redundant vectors.

My proposed solution is the SHM-VPR pipeline, which uses the WaSR segmentation as a guide for finding the horizon line, the section that separates the land/sea from the sky. WaSR allows us to extract this line for each image by traversing the x-axis of the generated segmentation map and finding the first y coordinate belonging to the land/sea class within each column, eventually forming an estimated horizon line.

By projecting these coordinates onto the images feature map, I can limit the sub-region extraction to a single set of windows across the x-axis, having the y-coordinate of each window be equal to the horizon line projection.

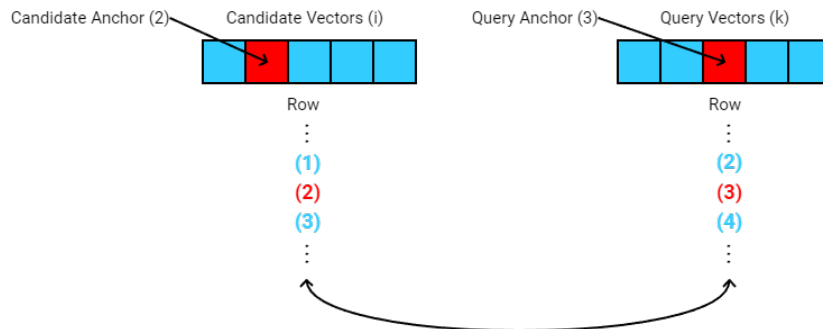


Figure 6.11: A simplified representation of the spatial matching stage for a row of query and retrieval vectors extracted via the SHM-VPR method, taking a pair of anchor points between the two, their surrounding vectors along the row should also match if the features are spatially consistent.

This produces a row of sub-regions rather than a grid, so the spatial matching stage is more streamlined and the amount of storage required for the spatial matching database is also reduced exponentially, this can be seen in Figure 6.11.

This method exploits the fact that across most shoreline imagery feature maps the horizon line is the most consistently highly activated feature, with most smaller land features being lost after multiple max-pooling operations due to the low resolution caused by distance from the camera, so instead of checking for spatial consistency across the whole image I limit it to the most structurally variant region.

## 6.6 Summary

For shoreline imagery, I find that SHM-VPR outperforms baseline SSM-VPR as it directly targets the inherent challenge of extracting the small percentage of salient information within the images while also trying to navigate the pipelines attention away from redundant features.

I recognize that this is not a universal state-of-the-art pipeline, but a domain-specific one, as the improvements made do not seem to translate to Symphony Lake, which makes sense as these inland images feature plenty of useful information across the whole image and at these smaller distances the horizon is of little relevance for navigational purposes.

The two augmented versions of SSM-VPR making use of unsupervised region proposal are functional but do not compete with the baseline version, suggesting that for now the brute-force sliding window approach is still the better method of region extraction.

## Chapter 7

# Semantic Segmentation based Knowledge priors for Waterborne Deep VPR

### Contents

---

7.1	Introduction . . . . .	122
7.2	Proposed Approach . . . . .	123
7.3	Dataset . . . . .	123
7.4	Architectures . . . . .	124
7.4.1	SSM-VPR (i.e. Baseline) . . . . .	124
7.4.2	SSM-VPR w/Segmentation Enhanced Feature Map (SEFM) [1] . . . . .	125
7.4.3	SSM-VPR w/Semantically Aware Local Descriptor Refinement (SALDR) [2] . . . . .	125
7.4.4	SHM-VPR Semantic Edge based SSM-VPR Stage 2 Spatial Descriptor Matching . . . . .	127
7.4.5	Semantically Aware SSM-VPR . . . . .	129
7.5	Comparative Analysis and Results . . . . .	129
7.5.1	Quantitative Analysis . . . . .	129
7.5.2	Qualitative Analysis . . . . .	133
7.6	Summary . . . . .	137

---

## 7.1 Introduction

In the previous chapter, I discussed several attempts at trying to optimize the Deep VPR local descriptor extraction process, which in the case of SSM-VPR [5] formed grids of local feature vectors via a simple sliding window approach.

I identified that one of the most notable features of waterborne imagery were large portions of low value semantic and spatial information (i.e. Sea and Sky), with the majority of useful information being limited to visible land. I attempted to use unsupervised region proposals [13, 127] to designate regions to convert into local descriptors for getting an initial set of candidate images from nearest neighbour search, however without training I found the sliding window maintained superior performance.

However, my modification of the SSM-VPR re-ranking stage, spatial matching, was able to achieve slight gains in terms of Precision-Recall over the original. This change was also motivated by my need to focus on relevant land-based features, as well as a known issue of SSM-VPR being storage requirements for stage 2 vectors. My novel approach used a segmentation map generated by the WaSR [37] network to create a binary mask representing the land class label, a semantic edge was then extracted from this mask and a set of coordinates were built along the edge to be projected onto the input image feature map, along which local descriptors for spatial matching were extracted.

This improved the re-ranking performance on the Plymouth Sound dataset, while reducing the number of vectors needed for SSM-VPR stage 2 and thus storage space exponentially. But, considering the added overhead and computational time introduced by incorporating segmentation the increase in performance was only marginal.

To address this, in this chapter I present a much more comprehensive overhaul of the SSM-VPR pipeline using WaSR as a semantic segmentation based knowledge prior, making use of this same output at all stages of the pipeline with my previous SHM-VPR method along with additional handpicked semantic methods. I show that these methods are able to provide a host of different benefits to the pipeline, while working seamlessly with one another to create a fused Semantically Aware SSM-VPR pipeline that balances the benefits of each.

In total I incorporate three methods into the pipeline individually before fusing them: the Naseer et al. [1] semantic feature map and therefore local descriptor enhancement method, the Hou et al. [2] semantically informed local descriptor refinement method and my SHM-VPR approach for the SSM-VPR re-ranking stage which limits spatial matching to a row of vectors extracted along a relevant semantic edge.

## 7.2 Proposed Approach

I propose a quantitative evaluation of five Deep VPR pipelines on the Plymouth Sound dataset. These include SSM-VPR (i.e. Baseline), SSM-VPR w/ Naseer et al. [1] for semantic descriptor enhancement, SSM-VPR w/Hou et al. [2] for stage 1 local descriptor refinement, SHM-VPR for semantic edge based stage 2 local descriptor spatial matching and a fused pipeline I dub Semantically Aware SSM-VPR which combines all additional semantic methods into a single pipeline.

Through implementation and evaluation of each pipeline I show how semantic segmentation based knowledge priors can be used to enhance several facets of Deep VPR performance, especially within the domain of waterborne imagery where feature for effective retrieval are largely contained within the semantic land label class.

Afterward, I show a more in depth qualitative comparison between the different semantic methods when integrated into the SSM-VPR pipeline, to try and gain a better understanding of just how these methods interact with standard CNN feature map outputs before the latter are then used to generate local descriptors for Deep VPR.

## 7.3 Dataset

The Plymouth Sound dataset consists of several traversals along the ares of its namesake captured by the IBM/Promare Mayflower Autonomous Ship (MAS). Traversals begin at Turn-chapel Wharf and cover differing areas of Plymouth Sound including Drake Island, Cawsand Bay, Rame and Whitsand Bay.

There are seven original runs in total taken between 30/03/2022 and 14/04/2022, so for evaluation I divide the overall image set into seven leave-one-out cross validation test folds. For some folds that stray especially far from Plymouth Sound such that their training folds do not have any matching ground truths for some queries, I only evaluate those queries that do have possible ground truths within said training folds according to a ground truth radius.

The dataset is quite challenging due to large variations in the distance of the camera to the visible shore as the MAS travels outwards into the Plymouth Sound, it also has more general viewpoint/feature changes for most locations based on the day recorded and is very sparse in terms of land features for performing place recognition. Average land pixel percentage across all images is around 5% according to WaSR, influenced heavily by the amount of images capturing open sea.

For more information, please refer back to Chapter 4 of this thesis.

## 7.4 Architectures

### 7.4.1 SSM-VPR (i.e. Baseline)

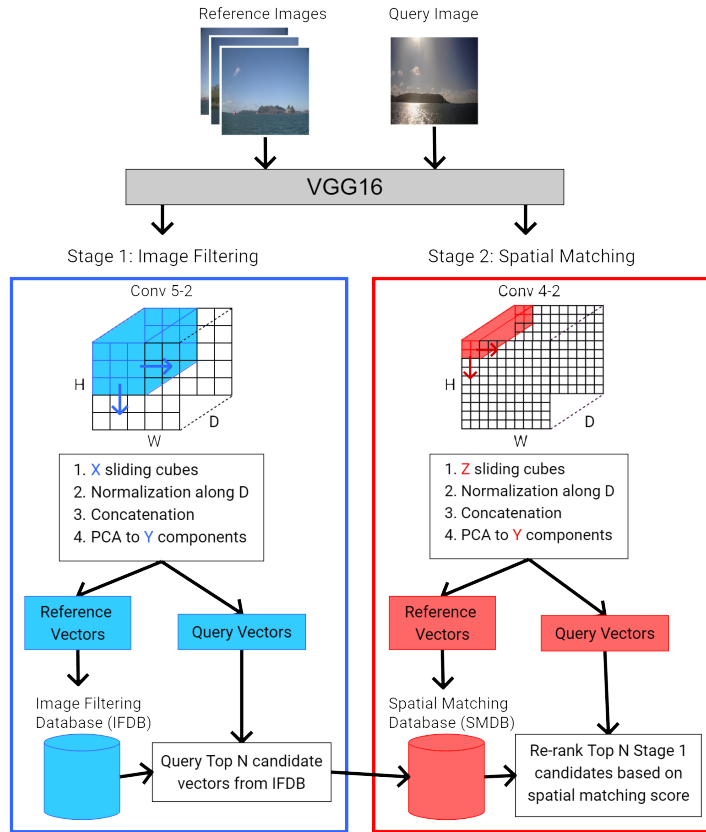


Figure 7.1: The pipeline of Baseline SSM-VPR, as described in [5].

Depicted in Figure 7.1 is the original SSM-VPR pipeline, for stage 1 I use the suggested resolution of  $224 \times 224$  for the images, producing a feature map of  $14 \times 14 \times 512$  over which I use a sliding window of  $9 \times 9$  for sub-region extraction. Resolution for stage 2 is  $448 \times 448$ , which was found to be more effective in the pipelines follow up paper [88], resulting in a feature map of  $56 \times 56 \times 512$  dimensions, the sliding window applied to this map has a dimension of  $3 \times 3$ .

I made one change concerning the extraction of sub-regions in both stages, I noticed that because the VGG16 backbone uses same padding for each convolutional layer the edges of



each feature map become highly activated, adding false edge features to the search databases. Therefore I limit the range of the sliding windows to avoid the edges of the feature maps.

#### **7.4.2 SSM-VPR w/Segmentation Enhanced Feature Map (SEFM) [1]**

This semantic enhancement method uses two versions of an input image, the original and a version masked according to a binary mask generated from a segmentation map label. These are then passed through a CNN and their output feature maps are aggregated to produce a single enriched feature map that promotes features falling within the binary mask. This can be applied to any Deep Learning task where certain classes of object are more desired, in my case I want to enhance areas belonging to the WaSR land class label.

I apply SEFM to SSM-VPR whenever an image is passed through its VGG16 backbone, including the building of both the Image Filtering and Spatial Matching Databases as well as during inference time.

SSM-VPR already provides a robust method of dimensionality reduction for CNN features during vectorization, dimensionally flattening them and applying PCA initialized on a random sample of training vectors at the start of database creation. For this reason I will not be making use of the sparse random projection used for reduction as was originally proposed for the segmentation enhanced feature map [1].

#### **7.4.3 SSM-VPR w/Semantically Aware Local Descriptor Refinement (SALDR) [2]**

In Deep VPR many pipelines make use of local descriptors based on image sub-regions, selecting these can be done through various methods, including fixed sliding windows [5] and off-the-shelf proposal algorithms [67, 179], however depending on how robust each method is, some of these regions may contain unwanted information so being able to refine these is useful for building an enriched local descriptor search space.

Hou et al. [2] propose a local descriptor refinement method that uses semantic segmentation knowledge priors, where each descriptor must pass a threshold of what percentage of pixels in their associated image sub-region are labelled with valid object classes for localization. This is highly beneficial as it ensures the search space is not inflated with uninformative local descriptors, subsequently optimizing the search space as a whole.

I apply this SALDR method to the SSM-VPR pipeline, once again using the WaSR land class label as the validity criteria. The amount of land pixels contained in most of the images

## 7. Semantic Segmentation based Knowledge priors for Waterborne Deep VPR

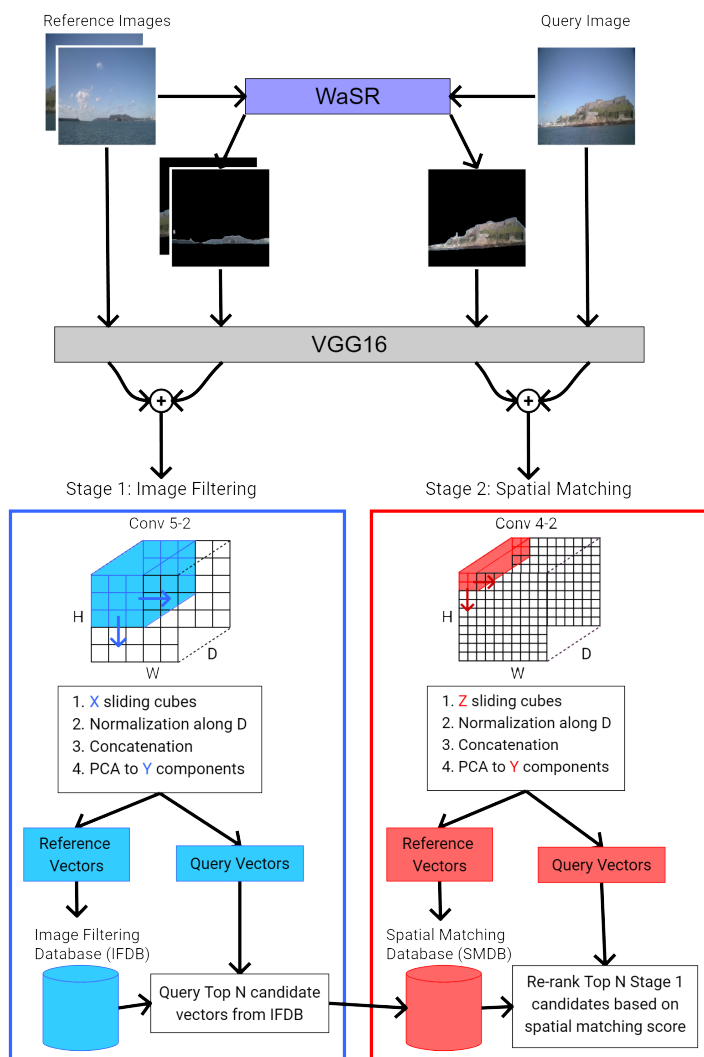


Figure 7.2: SSM-VPR with SEFM applied to all feature maps passed through VGG16 for stages 1 and 2. This applies to both training images (reference) and query images.

is already low, so I only set a small threshold of at least 5% land pixels in each local descriptor associated sub-region, as this was the average across the dataset as determined by WaSR.

Local descriptors associated with semantic regions that do not meet this threshold during database creation are withheld from the SSM-VPR Image Filtering Database (IFDB), during inference time local descriptors extracted from the query image that fail to meet the threshold are skipped over during query vector nearest-neighbour search. This prevents both the IFDB from being filled and the SSM-VPR image filtering stage histogram score from being influenced by vectors of low semantic-information.

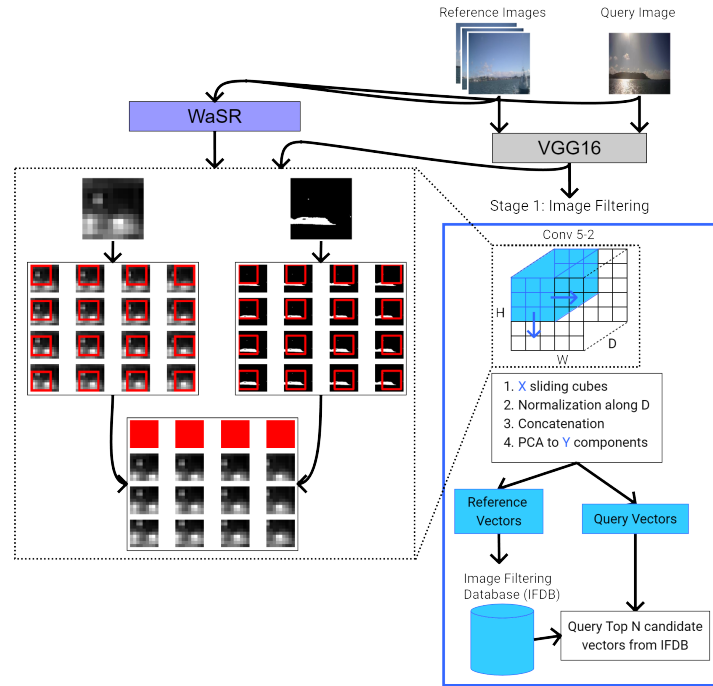


Figure 7.3: SSM-VPR with SALDR applied to the sliding window process of SSM-VPR stage 1. Feature map sub-regions designated by sliding window are projected onto a segmentation mask, if this area does not correspond to a minimum percentage of valid pixels it is discarded before moving on.

As a compromise, I only apply this to local descriptors from stage 1 of SSM-VPR, as stage 2, Spatial Matching, requires each example to have a fixed set of local descriptors that can then be reshaped into a grid for the spatial matching re-ranking algorithm.

#### 7.4.4 SHM-VPR Semantic Edge based SSM-VPR Stage 2 Spatial Descriptor Matching

I once again show results for my SHM-VPR method which shares similarities with WASABI [176], this only effects the Spatial Matching algorithm and local descriptor extraction.

Implementing the segmentation map post-processing stage in [176] is difficult for the image set as the minimum “blob size” of the semantic land object can vary based on distance. As such, I instead filter out all but the largest component belonging to the semantic land label, which leverages the fact that shorelines naturally form single, major land components. A con of this approach is that without additional segmentation methods to identify pixels belonging to boats, these objects can sometimes wrongly be identified as the largest land component, there are also some locations where two separate shore components can be seen, in which case only one will

## 7. Semantic Segmentation based Knowledge priors for Waterborne Deep VPR

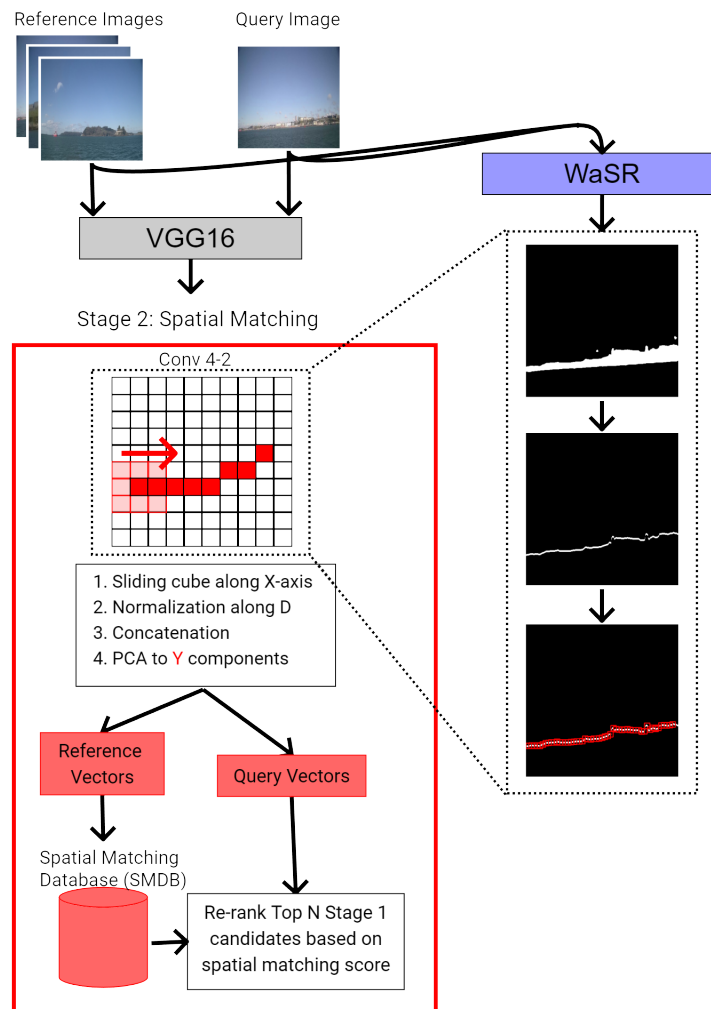


Figure 7.4: Our SHM-VPR alternative stage 2 method, using WaSR I extract a semantic edge belonging to the land class label, I then traverse the line to build a set of sliding window coordinates that can be projected back on to the stage 2 feature map. The sliding window goes across the x-axis, using the projected y-coordinate of each sampled point along the semantic edge.

be left unfiltered.

I extract a semantic edge for valid class labels (In my case just land), local descriptors are then extracted via a sliding window that moves along a projection of the semantic edge. In instances where one would have multiple valid class labels I would extract individual local descriptor sets for each individual class semantic edge, a spatial matching score can then be calculated for each set of local descriptors extracted from the same class label edge, adding up the score for an overall score between a Query and Retrieval Candidate image.

This process produces a single row of local descriptors per semantic class edge as opposed

to a generalised grid, which in the case where I only have one valid class label exponentially reduces the overall number of vectors in the SMDB, however, even with more valid classes storage is still massively reduced.

#### 7.4.5 Semantically Aware SSM-VPR

A fusion of pipelines 2-4 into a single comprehensive pipeline. Each individual enhancement works independently of one another using the same initial WaSR segmentation mask, introducing minimal overhead while keeping the overall methodology of Baseline SSM-VPR intact.

### 7.5 Comparative Analysis and Results

#### 7.5.1 Quantitative Analysis

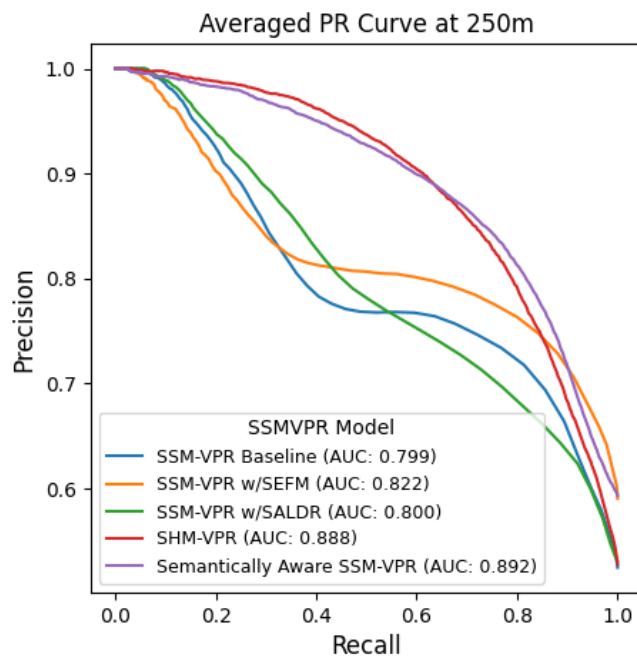


Figure 7.5: PR-Curves with AUC values for each pipeline. Note that only valid queries are included in the calculation, many waterborne images contain no information (i.e. Open Sea) and are therefore not valid. Queries are determined to be valid if their land pixel percentage from WaSR segmentation map is 5% or more.

Here I present Precision-Recall + AUC values, Recall@N and average inference time for each of the pipelines. For Precision-Recall, instead of using the mAP@1 metric for calculating

## 7. Semantic Segmentation based Knowledge priors for Waterborne Deep VPR

---

precision at for examples that fall within the descending absolute score interval as in previous works [5, 60, 61], I instead opt to calculate precision based on Noh et al.'s [91]'s modification of the  $\mu$ AP metric.

This enables consideration of average model precision beyond just the top re-ranked retrieval by including all retrievals whose absolute scores fall within the descending interval used to build the curve, with recall now representing the absolute number of retrievals normalized to range 0.0-1.0.

Figure 7.5 shows the  $\mu$ AP Precision-Recall + AUC values for each pipeline, and from this we can see a few things: Firstly, both SSM-VPR baseline and w/SEFM form similar curves to each other, with the latter being consistently higher, which is reflected in their AUC values. This is likely because the segmentation enhanced feature map (SEFM) promotes land features within images, making them more discriminative, but does not alter the workflow of SSM-VPR in any way once this input is taken.

SSM-VPR w/SALDR presents a middle-ground, having a smoother curve that outperforms the first two pipelines initially before underperforming as the minimum absolute score interval decreases, achieving an AUC value somewhere between SSM-VPR baseline and w/SEFM. This is likely because SALDR rejects local vectors without sufficient land information, meaning each individual vector used for matching in SSM-VPR stage 1 is likely of higher quality, but rejections means there are less to compare overall, which may dilute SSM-VPR's histogram scoring technique especially when images have many rejections.

Finally, both SHM-VPR and the fused Semantically Aware SSM-VPR pipeline present much improved curve shapes compared with the rest, giving them significantly higher AUC values as a result. We can determine from this that SHM-VPR is the most significant method for improving Precision over Recall, likely due to the area it spatially matches (the detected land edge) containing more discriminative features compared to the images as a whole.

This difference in curve shape and AUC values between SHM-VPR and baseline is far more significant compared to previous work [61], a clear impact of switching to the modified  $\mu$ AP metric. This suggests that when analysing all retrievals made per query rather than just those re-ranked to being the top retrieval, SHM-VPR's absolute/re-ranking score is able to distinguish between high to low precision retrievals much more gradually and effectively, as can be seen in its lack of any intermediate curve dips. By comparison, the curves produced by pipelines using the traditional SSM-VPR stage 2 core all show a large initial dip in precision between the top-most scoring retrievals and those that fall below before eventually stabilizing.

## 7. Semantic Segmentation based Knowledge priors for Waterborne Deep VPR

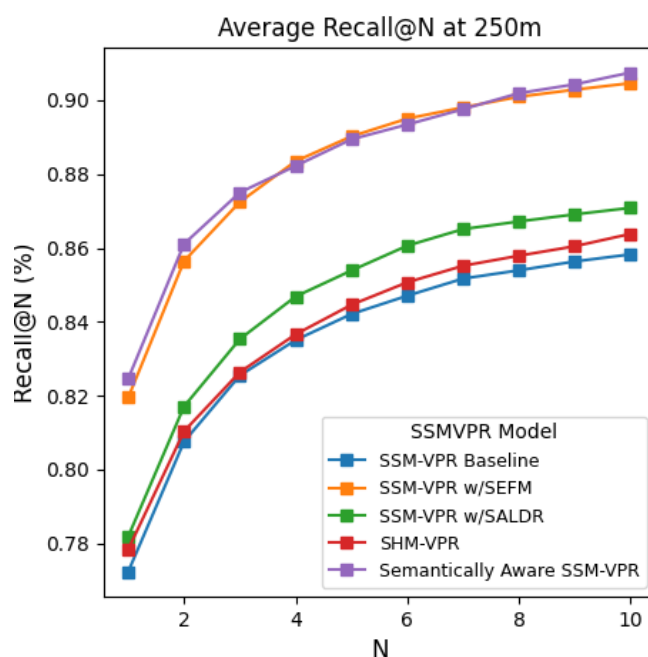


Figure 7.6: Recall@N percentage values for each pipeline. Only valid queries are included in the calculation.

Figure 7.6 shows the Recall@N metric from  $N=1$  to  $N=10$  for each pipeline, in other words the percentage of query images who have a true positive within the top- $N$  retrievals. Here the relationship between the pipelines is more pronounced compared to their Precision-Recall curves, beginning with baseline SSM-VPR and SHM-VPR, we see only a marginal difference in recall percentage across values for  $N$ , suggesting that although the SHM-VPR stage 2 score may be more effective for determining precision, in terms of overall recall both are similar in effectiveness.

Continuing in ascending order, SSM-VPR w/SALDR shows a consistent marginal improvement over both baseline SSM-VPR and SHM-VPR, suggesting that, although the PR curve results were somewhat inconclusive, the rejection of local vectors comprising mostly sea and sky has an overall benefit on the models ability to successfully recall.

The largest improvement is in the methods that incorporate the semantic feature map aggregation method, with both SSM-VPR w/SEFM and the Semantically Aware SSM-VPR pipeline showing similar increases in Recall@N across the board compared to all other methods. It would seem then that for some images, enhancing the land pixel features can make the difference between SSM-VPR recalling them, likely due to them being a small subset of the image

## 7. Semantic Segmentation based Knowledge priors for Waterborne Deep VPR

that rely on enhancement to become discriminative.

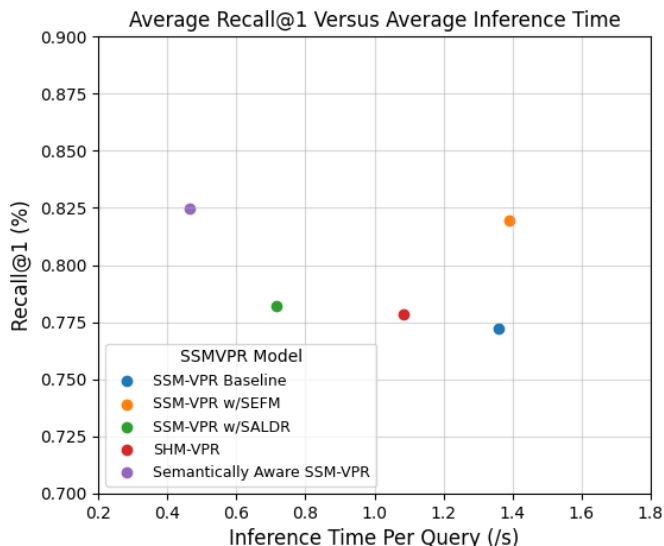


Figure 7.7: Average Inference Time per query in seconds versus Recall@1 for each pipeline. Only valid queries are included in the calculation.

Finally, in Figure 7.7 we can see the average inference time for a query versus Recall@1 for each pipeline, given that the pipeline is intended for real-time navigation being able to minimize inference time is important in terms of utility, applying the WaSR segmentation network to enable the individual semantically aware methods can clearly impact performance in terms of Precision-Recall (SHM-VPR) and Recall@N, but here we can see the trade-off in terms of computational time.

It is important to note that when building the retrieval image vector databases prior to any inference, pipelines making use of WaSR would predict and store the segmentation maps for the images, such that the process would not need to be repeated for retrieval images during inference.

In order of the labelled pipelines, we can see that baseline SSM-VPR took an average of just below 1.4s for each valid query, however surprisingly the introduction of WaSR segmentation prediction for SSM-VPR w/SEFM only results in a marginal increase closer to 1.4s exactly. Interestingly, SSM-VPR w/SALDR, maintaining a similar Recall@1 to baseline, was actually able to achieve a faster inference while still using WaSR to gain knowledge priors.

These improvements are a result of two factors: Firstly, the SALDR method filters out unwanted retrieval image vectors from the SSM-VPR stage 1 Image Filtering Dataset before any inference is done, reducing the search space. Secondly, for each query image, if during



stage 1 of SSM-VPR there are a number of sub-regions not suitable for vectorization, then the amount of IFDB searches for the query is reduced.

SHM-VPR also improved inference time compared to SSM-VPR w/SEFM but provided less improvement in the Recall@1 metric, likely due to the simplification of the stage 2 re-ranking algorithm to a 1D array of vectors rather than a 2D array for spatial matching.

The final fused model, Semantically Aware SSM-VPR, achieves the fastest inference time of all, a direct result of mixing the three semantic methods, as such the inference time has likely been reduced by both the SSM-VPR w/SALDR and SHM-VPR methods.

Taking all of these insights into account I make a series of conclusions for each of the individual semantically aware methods; Firstly, for SEFM, I achieve a raw boost in overall performance, improving Precision-Recall and Recall@N versus baseline SSM-VPR with a minimal inference time increase due to WaSR prediction. Secondly, for SALDR Precision-Recall is arguably smoother but overall only marginally different from baseline SSM-VPR, Recall@N sees a more clear yet still only marginal improvement and inference time sees a significant improvement. Thirdly, SHM-VPR significantly improves the Precision-Recall curve yet has seemingly little impact on the Recall@N compared to baseline SSM-VPR, while also improving inference time.

By combining these methods into the Semantically Aware SSM-VPR pipeline, I am able to balance these benefits and drawbacks in order to get the best of each; Precision-Recall is still comparable and even marginally better than SHM-VPR, Recall@N maintains the significant increase provided by the SEFM method, and the combination of SALDR and SHM-VPR reduces inference time even further.

### 7.5.2 Qualitative Analysis

Here I present a visual analysis of each semantic segmentation based method that I have introduced to the pipeline, including the effect they have on the feature maps and vectors generated from a set of example images. First, I will compare the SEFM method for semantic feature map generation with the regular CNN output:

In Figure 7.8 we see a basic diagram comparing a feature map generated from an input image using regular CNN and the SEFM segmentation method. With the latter incorporated, areas once effected by noise (i.e. the clouds to the left) are now suppressed compared to the regular CNN feature map and features associated with the land are more distinct.

Viewing a set of simplified examples in Figure 7.9, we can only perceive a subtle change on

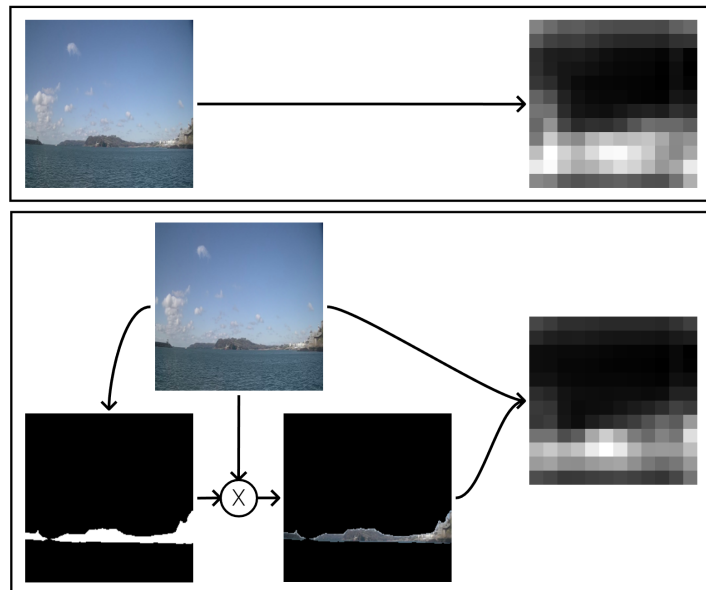


Figure 7.8: Top: A regular CNN feature map output for a given image. Bottom: A feature map produced using the SEFM method, for a given input a segmentation map is used to create a binary mask for the input, the original and masked inputs are both passed through the CNN and their feature maps are aggregated into a final output.

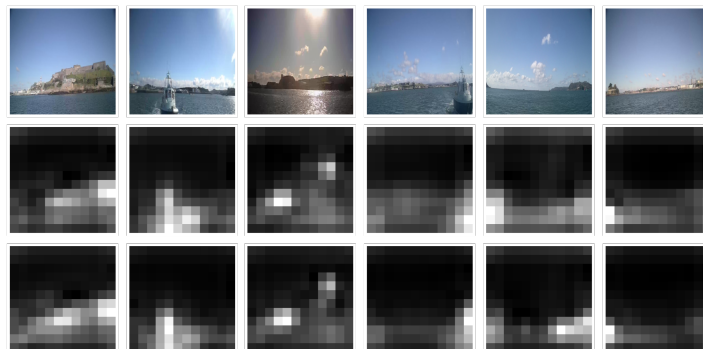


Figure 7.9: Top: Six input images examples. Middle: Regular feature maps generated by VGG16 Conv5\_2. Bottom: Feature maps generated by VGG16 Conv5\_2 using the SEFM segmentation method.

the surface, that being a reduction of noise within features associated with non-land pixels and a slight enhancement of those that are related to the land pixels, with this trend more clearly seen on the individual feature maps themselves.

With that in mind, we analyse the effect of the SALDR semantic feature vector filtering method on stage 1 of the SSM-VPR pipeline, which extracts feature vectors based on feature map sub-regions extracted via sliding window:

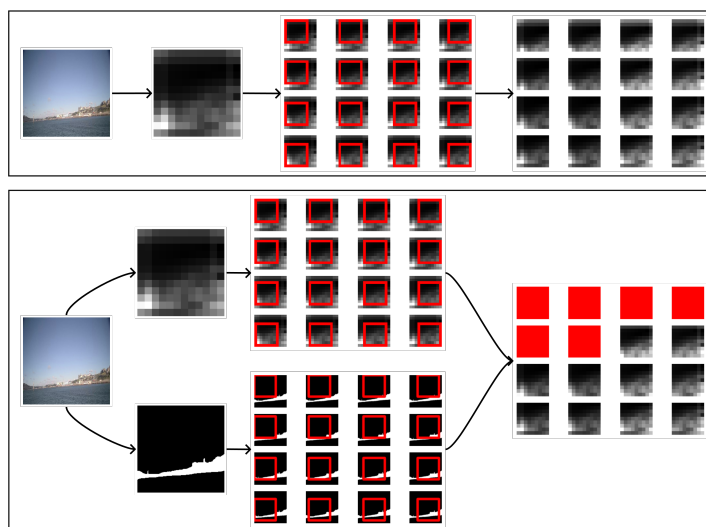


Figure 7.10: Top: An image produces a CNN feature map which is divided into a set of sub-regions via sliding window to be extracted and vectorized for nearest neighbour search as of a Deep VPR pipeline. Bottom: In addition to the feature map output, the image is also used to produce a segmentation map which is converted into a binary mask of valid/invalid class labels before semantic awareness is applied to filter valid regions.

In Figure 7.10 we see a comparison of CNN feature map sub-regions for local vectorization between a normal sliding window approach and the SALDR method, whereas the former simply extracts all sub-regions whose coordinates fall within the valid parameters during the windows stride, the latter projects the window at each step onto a binary mask representing valid class labels from a segmentation map output based on the input image. Whenever the amount of valid pixels within the projected window is too low, the corresponding sub-region of the feature map is rejected for local vector extraction. In my case, because the average land content in class-labelled pixel percentage across all Plymouth Sound images was 5%, I use this as the threshold.

The result of this, as can be seen in Figure 7.10, is that many feature map sub-regions linked to the upper portions of the image are rejected (shown by the red squares) as they corresponded to almost completely non-valid pixel label features, likely containing unwanted residual feature values. More of examples of the SALDR method being applied to Plymouth Sound can be seen in Figure 7.11.

Finally, I present the results of applying the SHM-VPR method for SSM-VPR stage 2 in Figure 7.12, which builds a single row of windows from which to extract local feature map sub-regions for vectorization by following the edge of a binary mask generated from a valid

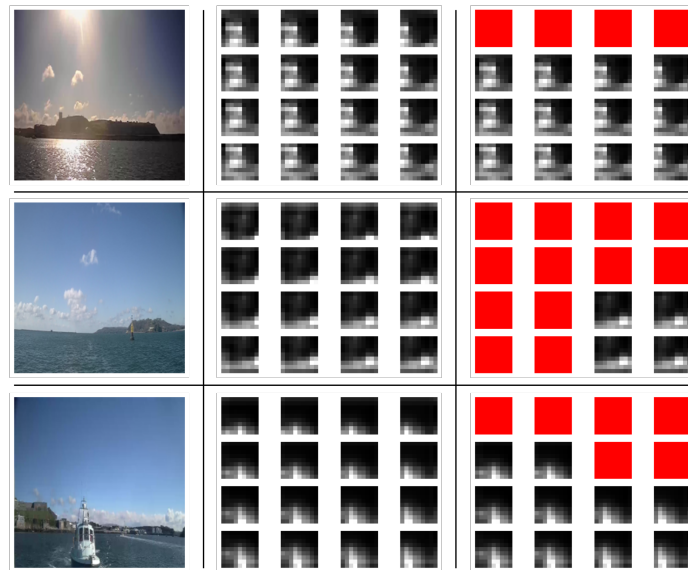


Figure 7.11: First Column: Example image inputs. Second Column: Resulting CNN Feature map sub-regions selected for local vectorization via sliding window. Third Column: Same as second column but with sub-regions rejected by SALDR depicted as red squares.

semantic class label.

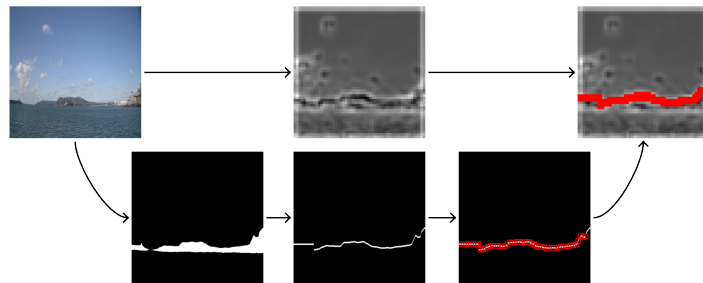


Figure 7.12: The SHM-VPR alternative to SSM-VPR stage 2, instead of producing local vectors from a small sliding window across a VGG16 Conv4\_2 feature map, we first take a binary mask based on a particular valid semantic class label and extract its upper edge. A row of windows are then localized along this edge and projected onto the feature map for local vector extraction.

By limiting the spatial matching of SSM-VPR stage 2 to a row of vectors built up of geometric features along the semantic edge, I firstly remove the need to store an exponentially larger grid of vectors per image massively reducing storage space, secondly, the spatial matching algorithm is now much more focused on the most geometrically significant feature of most waterborne imagery, the horizon or skyline.

As can be seen in Figure 7.13, the method is effective at aligning windows along the main

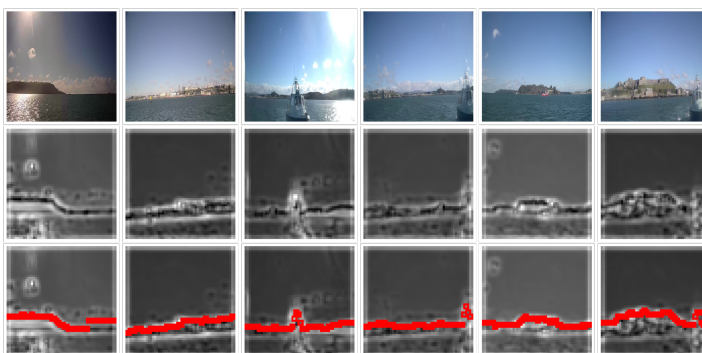


Figure 7.13: Top: Six input images examples. Middle: CNN feature maps generated by VGG16 Conv4\_2 for SSM-VPR stage 2. Bottom: Windows projected along a semantic land edge generated from each image to be used by SHM-VPR stage 2.

geometric feature seen within the feature maps, although there are some teething issues. In the first column you can see a rise in the position of the windows to the right where there is no longer any land in the image, this is because in columns of the segmentation mask where there was no land to draw an edge position from I simply used the mean edge position at that point, in future I may simply choose this to be the minimum instead.

A more pressing issue is the interference caused by boats, in the third column of Figure 7.13 we can see the clearest example of this. WaSR only classifies land, sea and sky and was originally intended for obstacle detection, as boats are one of the most common of these, WaSR brings them under the umbrella of the land class such that the system can be programmed to simply avoid objects belonging to this label. Unfortunately for us this means that boats can disrupt the semantic land edge within the segmentation map, which in turn disrupts my SHM-VPR method. The best solution in future would be to train an extended WaSR based network to classify boats as a separate class label such that we can filter them from semantic operations.

## 7.6 Summary

Building upon my implementation and evaluation of the SHM-VPR model [61], I find that a more comprehensive Semantically Aware SSM-VPR pipeline making use of three methods taking a single segmentation prediction map as input [1, 2, 61] is able to outperform SSM-VPR and SHM-VPR in terms of both Precision-Recall according to absolute retrieval score and average recall across all queries given N candidates, while keeping the increase in inference time brought on by semantic prediction network WaSR to a minimum.

## *7. Semantic Segmentation based Knowledge priors for Waterborne Deep VPR*

---

I believe the two semantic segmentation based enhancement methods that I have hand-picked, alongside the third method from my previous work, is able to strike an excellent balance between all metrics showcased within this work when applied to the task of Deep VPR for Waterborne Domains.

Given the nature of imagery with the Waterborne Domain, more specifically that of longer-range image captures from a large shoreline based environment depicted in the Plymouth Sound dataset, I have shown that semantic segmentation based enhancements for Deep VPR are effective for tackling the numerous challenges inherent to these images, including the promotion of features within the land labelled minority of many waterborne images, discarding of non-informative feature vectors generated from empty sea or sky labelled regions and a focus on spatially matching geometric feature vectors along a designated semantic edge line.

## Chapter 8

# Measuring User Reliance and Technology Dominance in Waterborne Deep VPR With Human-Centered AI

### Contents

---

8.1	Introduction . . . . .	<b>140</b>
8.2	Method . . . . .	<b>141</b>
8.2.1	Architectures . . . . .	141
8.2.2	Dataset . . . . .	143
8.2.3	User Study . . . . .	144
8.3	Comparative Analysis and Results . . . . .	<b>151</b>
8.3.1	Quantitative Analysis . . . . .	151
8.3.2	Qualitative Analysis . . . . .	156
8.4	Conclusions . . . . .	<b>159</b>

---

## 8.1 Introduction

So far my bodies of work have tackled the technological side of Deep VPR, testing viability, application and subsequently improving objective performance, for my final body of work I shift to the human evaluation of Deep VPR for autonomous navigation.

Deep Visual Place Recognition (Deep VPR) is usually employed by fully autonomous vehicles to tackle “error drift”, a problem whereby small localization errors made by an autonomous system add up to a serious error over time. Deep VPR can identify previous locations captured via on-board cameras, using this knowledge to verify its current localization versus the previous localization, then, if an error is detected, the system can attempt to correct error, a concept named “Loop Closure” [103]. However, VPR also has value as a decision-support system for helping human users complete the much broader task of identifying/verifying their global position [180].

It is clear that the idea of a fully autonomous navigation system can incite feelings of anxiety and mistrust in potential users [181], due to factors such as lack of human agency and legal issues in case of accidents [108], dissuading the adoption of these systems. For decision-support systems adoption has been more successful, especially in areas where human error could have serious consequences such as medical work [182], however this does not necessarily correlate with their actual effectiveness towards supporting the given task, as user positive and negative biases towards the system can lead to extreme cases where users allow the AI full control or never engage with the AI respectively, given it is the case where both the user and the AI can make both correct and incorrect decisions, often across different cases, this would result in sub-optimal performance as opposed to a mixed approach where the two can adjust and combine their decision making with each other [105].

A major factor behind whether or not a user will adopt and subsequently rely upon an AI system is trust [111]. One of the major works related to measuring trust between humans and AI is Jian et al. [10], who subsequently provided the most recognized and sourced list of suggested likert scales for measuring trust in the context of an AI-based user study.

Many researchers point out the black-box nature of AI creating a lack of user trust (i.e. “Why does the machine make this decision?”), making users reliance on such systems quite low. Two Human-Centered AI approaches to solving the black-box problem are Explainable AI (XAI) and Human-in-the-Loop techniques, the former revealing model decision-making in a human interpretable way and the latter allowing the user to be involved in decision-making directly.



## **8.2 Method**

We propose a user study to measure the change in User Reliance and Technology Dominance in Deep VPR for decision support with three levels of Human-Centered AI; None, Explainable AI and Human-in-the-Loop. I carry this out in a Waterborne context to be in-keeping with my previous research as well as to provide some novel insight into the autonomous vessel domain.

Reliance, in this context, is the willingness of users to hand over control or agency to the AI system when using it for decision-support [104]. I believe it is a word closely related with trust and confidence although whereas trust is a very wide-ranging term, reliance is much more specific and thus easier to properly analyse.

Technology Dominance is an even more specific term related to user reliance on AI, it is a term defined by Cabitza et al [105], and can be described as how much a piece of technology dominates a human users decision making process, for example a user allowing the technology to make all decisions will result in high technology dominance.

Firstly, the users will interact with a basecase version of the Deep VPR system, where they are presented with an input and output image with no explanation and no Human-Centered AI methods (i.e. Black Box), acting as the control results.

For XAI, I propose the reintroduction the novel image saliency technique introduced in Chapter 5 to show end users what visual features are deemed important in the Deep VPR retrieval based on similarity to the input image. This consists of placing a saliency map onto the output image whose intention is to highlight matching features between it and the input.

Then, for human-in-the-loop, I introduce an active learning component where the user can choose to overwrite the land segmentation mask proposed by the WaSR model integrated into the Deep VPR pipeline in Chapter 6 and 7, giving the user more control over this intermediate output stage which is known to make mistakes on its own.

### **8.2.1 Architectures**

#### **8.2.1.1 Basecase**

For waterborne Deep VPR, I use a novel modification of the SSM-VPR [5] architecture which incorporates WaSR [37] land, sea and sky segmentation at multiple stages of the pipeline which I have shown in previous research can produce optimal retrieval outputs within my specific image domain.

## 8. Measuring User Reliance and Technology Dominance in Waterborne Deep VPR With Human-Centered AI

---

Users are expected to view an input and retrieval image pair from this model along with the position of the output on a map, then, given this information, decide if they would like to use this output to localize themselves and enter their feelings of reliance towards the system on a per question basis, through a set of scales that I will discuss in the user study subsection. To aid the user in this process, a circle containing (But not centered on) the ground truth is shown to simulate how a person may have an approximation of their position but not an exact fix 8.1. Scenario 1 will provide a relatively small ground truth circle whereas 2 presents a larger circle and 3 presents no circle.

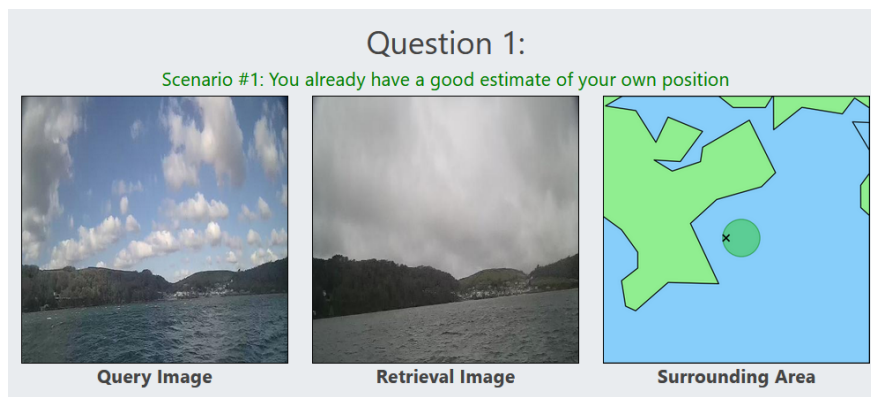


Figure 8.1: An example of how a user will see the Basecase Deep VPR image input and output during the survey

Implementation details are kept hidden so as to maintain the black box nature of the AI, they are given only a brief summary of how a Deep VPR system works in general before being shown various input images and corresponding retrievals.

### 8.2.1.2 Explainable AI

For Explainable AI (XAI), I use my novel Image Saliency method for Deep VPR based on Score-CAM [26], this is typically used for explaining image classification tasks, such as highlighting a dog to explain why an AI classifies an image as “dog”, but for Deep VPR, I have restructured the process so that it outputs a heatmap for the retrieval image where highlighted areas indicate features that are similar to the input image.

As for what is expected of the user everything outlined in the Basecase version remains the same, but there will be a third image shown that is the same output image with a saliency map overlaid to indicate why it was matched to the input, giving the user an explanation of the AI’s decision making 8.2.

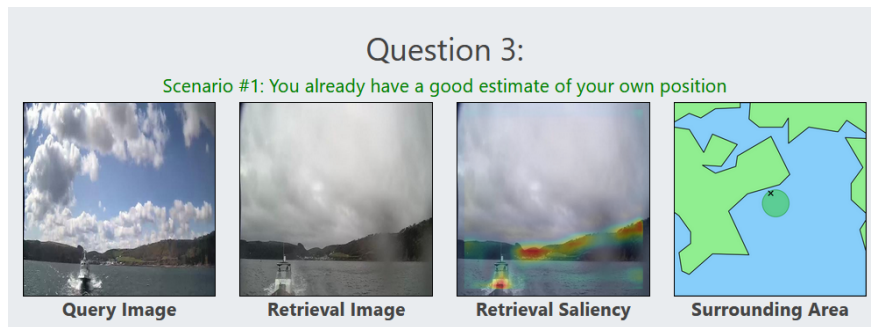


Figure 8.2: An example of how a user will see the Deep VPR image input, output and Image Saliency heatmaps during the survey

As the users are not expected to know what I mean by “Image Saliency” or “Explainable AI”, I add a section to the summary section (i.e. Before the experiment begins proper) including what these two things are and how to interpret the saliency map output.

### 8.2.1.3 Human-in-the-Loop

My human-in-the-loop method takes note from active learning methods, specifically the idea of using semantic segmentation as a suggestion for annotation, where users are provided with an initial labelling of relevant areas in an image and if there are any errors, they can correct it manually [170].

Although segmentation is an effective method that has worked well to improve the Deep VPR pipeline on waterborne imagery, it is not perfect and in many instances could be quickly corrected by human input. As such, before an input and retrieval image are shown to the user I show them the segmentation mask 8.3 for the relevant land portion overlaid onto the input and allow them to either leave it as is, or produce a manual mask which the pipeline can then use instead.

Once again, I do not explain the specifics of how this mask feeds into the effectiveness of the pipeline, but I do inform the user that informing the model of where the land is located improves model output as well as a brief description of human-in-the-loop interaction.

## 8.2.2 Dataset

The Plymouth Sound dataset consists of several traversals along the ares of its namesake captured by the IBM/Promare Mayflower Autonomous Ship (MAS). Traversals begin at Turn-

## 8. Measuring User Reliance and Technology Dominance in Waterborne Deep VPR With Human-Centered AI

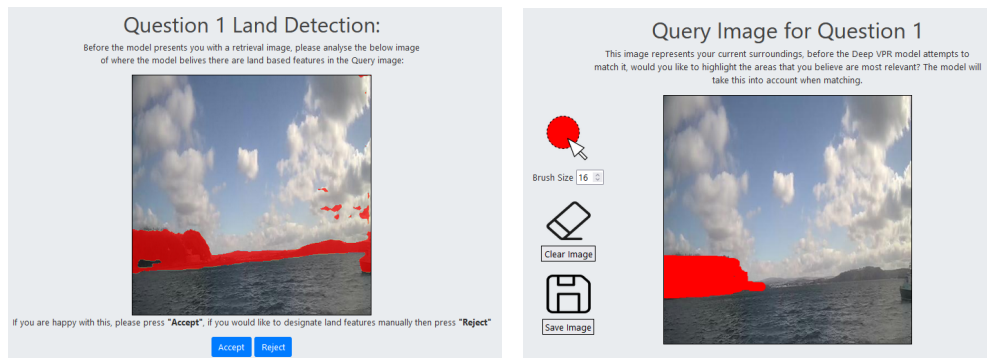


Figure 8.3: Left: An example interaction prompt from the human-in-the-loop survey, shows WaSR land segmentation in red. Right: If the user chooses ‘Reject’ on the left page, they are taken to a rudimentary image painting page where a new segmentation mask is created by highlighting parts in red.

chapel Wharf and cover differing areas of Plymouth Sound including Drake Island, Cawsand Bay, Rame and Whitsand Bay.

There are seven original runs in total taken between 30/03/2022 and 14/04/2022, so for evaluation I divide the overall image set into seven leave-one-out cross validation test folds. For some folds that stray especially far from Plymouth Sound such that their training folds do not have any matching ground truths for some queries, I only evaluate those queries that do have possible ground truths within said training folds according to a ground truth radius.

The dataset is quite challenging due to large variations in the distance of the camera to the visible shore as the MAS travels outwards into the Plymouth Sound, it also has more general viewpoint/feature changes for most locations based on the day recorded and is very sparse in terms of land features for performing place recognition. Average land pixel percentage across all images is around 5% according to WaSR, influenced heavily by the amount of images capturing open sea.

### 8.2.3 User Study

#### 8.2.3.1 Research Questions

With my goal of measuring Technology Dominance and User Reliance in the Deep VPR architecture on a Waterborne Plymouth Sound Dataset for the architecture at basecase, with explainable image saliency and human-in-the-loop capability, each of which shares a set of theoretical scenarios the user may find themselves in, I propose the following research questions:

**RQ1:** How is Technology Dominance affected by the current navigational scenario?

**RQ2:** How is Technology Dominance affected by the Human-Centered AI methods?

**RQ3:** How is User Reliance affected by the current navigational scenario?

**RQ4:** How is User Reliance affected by the image retrieval quality?

**RQ5:** How is User Reliance affected by Human-Centered AI methods?

These questions are essentially asking how each of the independent variables, Human-Centered AI methods, Scenario and Retrieval Quality effects the two dependent variables, User Reliance and Technology Dominance.

### 8.2.3.2 Design

My study is split into three arms, each of which represents one of the three levels of Human-Centered AI: Basecase (None), Explainable AI and Human-in-the-Loop, which forms the first independent variable, *Human-Centered AI methods*.

The study follows a common format across the three arms, users are presented with a series of 18 questions displaying an input and retrieval image from the Deep VPR model along with a map depicting the surrounding area as well as a black “X” marker for where the output image was captured.

To gather user results on outputs of varying quality or closeness to ground truth, I sample questions in such a way that there is an even amount of Optimal True Positives, Sub-Optimal True Positives and False Positives.

For clarity, the Deep VPR model always collects a number N retrieval candidates for a given input, being ranked in order of similarity, therefore Optimal True Positives are retrievals that are both correct (i.e. within range of the input) and were ranked at the top of the candidate list, whereas Sub-Optimal True Positives are those who were the lowest ranked correct retrieval and, finally, False Positives are simply incorrect retrievals. Questions belonging to these groups are assigned a label and form the second independent variable, *Retrieval Quality*.

In addition, as the user progresses through the survey, questions will be made a part of one of three scenarios, which depict users level of knowledge towards their ground truth position in the form of an indicator on the map. These include: The User has a good estimate of their Ground-Truth position, the user has a medium estimate and the user has no estimate. This forms the third independent variable, **Scenario**.

## 8. *Measuring User Reliance and Technology Dominance in Waterborne Deep VPR With Human-Centered AI*

---

Depending on the *Scenario*, the map image will contain one of the following elements: A small, green circle containing the ground-truth position of the query image, a larger yellow circle serving an identical purpose and no circle at all, with the position of the retrieval now being marked with a red “X” to better communicate that the user is within scenario 3.

I chose to distribute these scenarios in such a way that the user begins in Scenario 1, good knowledge of their position, then progresses to Scenarios 2 and 3 respectively. As such, 6 out of 18 questions will belong to each of these scenarios, making sure users encounter an even number of each.

To ensure good coverage of each individual question in the user study, I initialize a pool of 180 possible questions. Given that each study has 50 participants who will encounter 18 of these questions, this means 10% percent of the overall question pool will be covered each time, meaning each individual question will on average be covered by  $50/10 = 5$  participants.

In addition, to ensure questions belonging to each combination of the **Retrieval Quality** and **Scenario** variables receive even coverage, the pool of 180 questions is first divided into groups of 60 questions belonging to each *Retrieval Quality* label, then further divided into three sub-groups of 20, each representing one of the three *Scenario* labels.

In terms of data collection, the user will be asked to fill in 8 fields total, most of these feature a 7 point likert scale where 1 will be equivalent to saying “Not At All” and 7 will be equivalent to “Extremely”. The first two fields, which ask “*Would you accept the systems suggestion?*” and “*How confident are you in your personal decision making?*”, differ from the others in that I use them to calculate participants reliance on the Deep VPR model during their decision making process using two technology dominance metrics proposed by Cabitza et al. [105], Automation Bias and Detrimental Algorithmic Aversion.

These metrics act as odds ratios with the former being the ratio of likelihood of automation bias (detrimental over-reliance) and beneficial algorithmic aversion (beneficial self-reliance), as such, values over one indicate the AI may be inducing a negative outcome on the users decision making.

The Detrimental Algorithmic Aversion ratio on the other hand acts as an inverse, representing the likelihood of conservatism bias (detrimental self-reliance) and algorithmic appreciation (associated with beneficial over-reliance), as such values over 1 indicate that the AI fails to induce a positive outcome due to the user ignoring them.

The remaining fields are all used for determining positive and negative aspects of user trust towards the Deep VPR model during each question, as trust is often associated with

8. *Measuring User Reliance and Technology Dominance in Waterborne Deep VPR With Human-Centered AI*

reliance and these scales, made for measuring human-machine trust, come from the well cited Jian et al. [10] paper, each of which was made to represent word clusters associated with trust/distrust (See Figure 8.4). For each scale I adopt from Jian et al. I also adopt its ideological opposite, sticking to those that are most relevant to the study while also avoiding scales that have significant overlap with one another.

Item	Words Groups from Cluster Analysis
The system is deceptive	Deception Lie Falsity Betray Misleading Phony Cheat
The system behaves in an underhanded manner	Sneaky Steal
I am suspicious of the system's intent, action, or output	Mistrust Suspicion Distrust
I am wary of the system	Beware
The system's action will have a harmful or injurious outcome	Cruel Harm
I am confident in the system	Assurance Confidence
The system provides security	Security
The system has integrity	Honor Integrity
The system is dependable	Fidelity Loyalty
The system is reliable	Honesty Promise Reliability Trustworthy Friendship Love
I can trust the system	Entrust
I am familiar with the system	Familiarity

Figure 8.4: From Jian et al. [10], "Trust scale items for human-machine trust and the corresponding cluster of trust related words on which they were based"

From Figure 8.4, I use items 12 and 3 as the first pair of opposites, with item 12 representing the word "entrust" and 3 representing "mistrust", "suspicion" and "distrust". The second pair of opposites I propose are items 10 and 1 with the former representing "honesty", "promise" etc. and the latter representing "deception", "lie" etc.. The final pair I propose are items 8 and 5, as 8 represents "honor" and "integrity" whereas 5 represents "cruel" and "harm".

The reason I choose not to use the other items presented are as follows; Word groups belonging to items 2, 4 and 9 could be argued as synonyms of those represented by items 1, 3 and 10 respectively, item 6 overlaps with the field on users personal decision making confidence, item 7's meaning may be difficult to interpret for participants (i.e. What does security mean in this instance?) and item 12 does not make sense in a context where the user is using the system for the first time.

The implementation of these questions is depicted in Figure 8.5, a screenshot taken from one of the Basecase/XAI survey question pages.

### **8.2.3.3 Deployment Stage**

Firstly, I deploy the study on the survey website Prolific, which acts as a self-service data collection platform which links online participants to the survey, via external web-link. Each participant is paid an ethical hourly rate of £10.54 [183] for completing the study, with the Basecase and Explainable AI versions taking 30 minutes on average for a total of £5.27 per participant. The Human-in-the-Loop interactivity version takes more time to complete on average so I set it to 40 minutes and offer £7.03 per participant. Each human-centered variant of the study will be filled out by independently sampled group of 50 anonymous participants each, which will include anyone with access to a Prolific account.

My requirements of Prolific participants included being located in the UK, being 18 or over, having an equal 50/50 male to female split and having an approval rating of at least 90%. To make sure all participants were properly engaged in the study, I inserted a series of simple attention questions after the explanation of a question scenario which involves ticking two statements about the current scenario. Participants who consistently fail these attention questions will have their results excluded.

### **8.2.3.4 Qualitative Stage**

In addition, I carry out a qualitative study whereby I send a link to each of the study arms to 5 staff members of the MSubs Ltd for them to complete, I then carry out 30 minute structured interview with the participants to collect their thoughts on using the Deep VPR model under different levels of interactivity.

These requirements are in-keeping with similar studies into human-AI interaction in terms of diversity and quality of responses [184–186].



## 8. Measuring User Reliance and Technology Dominance in Waterborne Deep VPR With Human-Centered AI

The figure displays two screenshots of a survey interface titled "Testing User Confidence in Deep VPR".

**Left Screenshot (Question 1):** The scenario is "Scenario #1: You already have a good estimate of your own position". It shows a "Query Image" (a landscape with a boat), a "Retrieval Image" (a similar landscape), and a "Surrounding Area" map. Below the images, it asks "Would you accept the systems suggestion?" with "Accept" and "Reject" buttons. It then asks "Are you confident in your personal decision making?" with a 7-point Likert scale. A note states "(Note: Not At All=1; Extremely=7)". Below this is a list of statements for evaluating trust, each with a 7-point Likert scale: "I can trust the system", "The system is reliable", "The system has integrity", "I am suspicious of the system's output", "The system is deceptive", and "The system's output will have a harmful outcome". A "Submit" button is at the bottom.

**Right Screenshot (Question 3):** The scenario is "Scenario #1: You already have a good estimate of your own position". It shows a "Query Image", a "Retrieval Image", a "Retrieval Saliency" image (a landscape with a boat and a red circle), and a "Surrounding Area" map. Below the images, it asks "Would you accept the systems suggestion?" with "Accept" and "Reject" buttons. It then asks "Are you confident in your personal decision making?" with a 7-point Likert scale. A note states "(Note: Not At All=1; Extremely=7)". Below this is a list of statements for evaluating trust, each with a 7-point Likert scale: "I can trust the system", "The system is reliable", "The system has integrity", "I am suspicious of the system's output", "The system is deceptive", and "The system's output will have a harmful outcome". A "Submit" button is at the bottom.

Figure 8.5: Left: An example question from the Basecase Survey. Right: An example question from the XAI survey

The second deployment consists of 5 members of staff from MSubs Ltd, a company that specializes in manned and unmanned submersible vehicle deployment and manufacture, these staff members regularly pilot or perform maintenance on sea vessels as part of their work.

As such, each user is well-versed in the task of mapping maritime regions as well as having good knowledge on maritime navigation. Most of these participants do not regularly make use of AI models in their workspace, although they often work alongside Marine AI Ltd., who are a software company linked with MSubs focused exclusively on autonomous vessels using AI, so some users will likely have some knowledge based on this.

This makes them ideal targets for my research, as I want to gain insight into end users (maritime navigators) reliance on Deep VPR for decision-support.

### 8.2.3.5 Interview

As part qualitative study stage an additional 30 minute interview with each of the individual MSubs staff members is carried out, which are open-ended semi-structured interviews.

Interviewees will be asked the following questions:

*Q1: Overall, what were your feelings towards using Deep VPR for decision-support in this context (i.e. Getting an approximate position without GNSS)?*

*Q2: As the scenarios indicated less knowledge of your position, did you find that the decision-support dynamic changed?*

*Q3: When you encountered an incorrect suggestion by the AI, did your attitude towards the following suggestions change?*

*Q4: For the saliency (Explainable AI) survey, did you find the heatmap applied to the retrieval an effective means of communicating similarity to the input?*

*Q5: For the human in the loop survey, how often did you intervene in the models decision making? Did your attitude towards the AI change after intervening?*

The questions were designed following data collection and analysis of the online user study, as such I took the opportunity to design them such that I could get more insight into those results.

As such, each one uses clear and specific language in order to gain deeper understandings compared to the online survey, while keeping language neutral as to not provoke specific answers. The questions are ordered in such a way that earlier ones should not have an impact on later questions.

Question 1 is a broad, general question designed to extract the users thoughts on Deep VPR for decision-support as a whole.

Each following question is linked to the users thoughts on one of the independent variables, while keeping the language used clear and understandable, for example Question 2 is linked to the scenario aspect and attempts to derive further information by motivating the respondents to give thoughts on how it changed their approach to the AI.

Question 3 is linked to the Retrieval Quality aspect, it is specifically designed to gather information on how negative outputs can effect users attitudes in the long-term.

Question 4 is linked to Explainable AI and is aimed at judging it's perceived effectiveness from the point-of-view of the experts.

Finally, Question 5 is designed to investigate how often the experts used the Human in the Loop aspect and if it had a positive after effect on their opinion of the AI.

Having 30 minutes with each participant gives us 5 minutes to discuss each individual question.

#### **8.2.3.6 Ethics and Consent**

Both the User Study and Interview branches of this work were approved of by the Swansea University Faculty of Science and Engineering ethics committee. Letters of approval can be found with Research Ethics Approval Number 2 2025 8625 12673 and 2 2025 13329 12753 respectively.

At the beginning of the survey, a standard Swansea University Participant Consent Form was provided to the users to sign, the users could only continue on to the survey if this was completed, otherwise they would be sent to the end with no data collected.

### **8.3 Comparative Analysis and Results**

#### **8.3.1 Quantitative Analysis**

For research questions (**RQ1** and **RQ2**) relating to Technology Dominance, I perform quantitative analysis based on the results of my online survey by plotting box plots of user answers for Fields 3-8 and by performing a Multivariate analysis of variance (MANOVA) test to determine statistical significance.

I measure the results of MANOVA by analysing the P value of both Wilks' lambda and Pillai's trace, the former measure the difference in group means for a set of dependent variables given some independent variable and the latter measures the variance in the dependent variables caused by the independent variable. A Wilk's lambda value near 0 indicates that groups are well separated and a Pillai's trace value near 1 indicates strong evidence for the results of MANOVA. The P value is shown in the Pr > F field, a value near 0 indicates statistical significance

The dependent variables are the Automation Bias and Detrimental Algorithmic Aversion ratios for each user across two out of three of the independent variables: *Scenario* and *Human-Centered AI methods*.

*Retrieval Quality* was not used as an independent variable for this analysis as it directly ties in with the calculation of these metrics (i.e. Retrieval Quality of 3 = False Positive) so separating samples into groups according to these values would make calculation impossible.

**RQ1: How is Technology Dominance affected by Scenario?:**

8. *Measuring User Reliance and Technology Dominance in Waterborne Deep VPR With Human-Centered AI*

Scenario	Value	Num DF	Den DF	F Value	Pr > F
Wilks' lambda	0.6159	2.0000	445.0000	138.7529	0.0000
Pillai's trace	0.3841	2.0000	445.0000	138.7529	0.0000

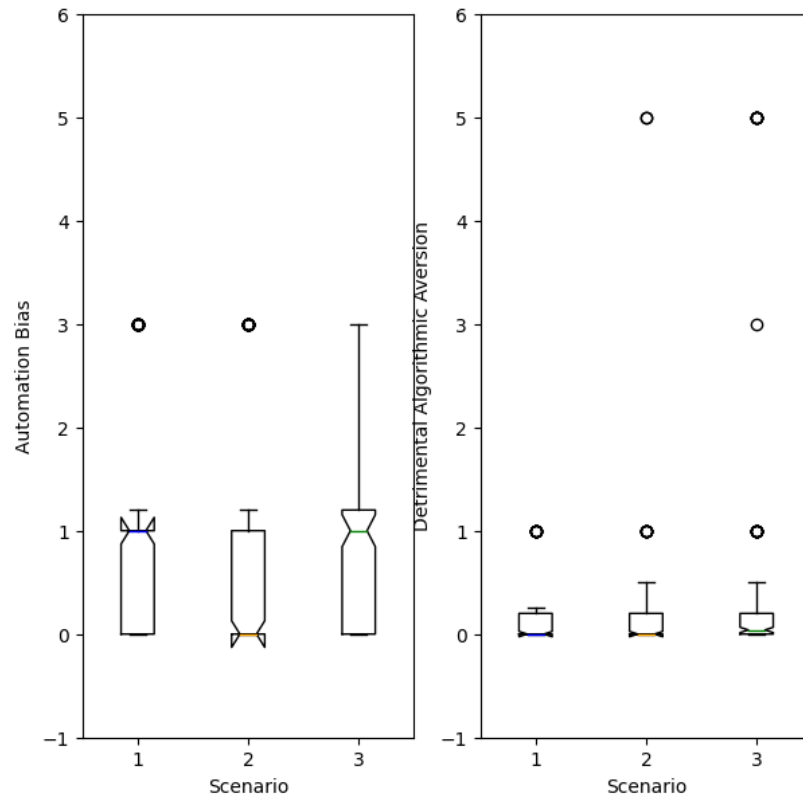


Figure 8.6: Left: Box Plots for Automation Bias ratio for all users across Scenarios 1-3 Right: Box Plots for Detrimental Algorithmic Aversion ratio for all users across Scenarios 1-3.

The results for Technology Dominance over Scenarios show significant difference with  $Pr > F$  of 0, but with a high Wilk's lambda value of 0.6159 and a low Pillai's trace of 0.3841. We can see in the box plots for Automation Bias and Detrimental Algorithmic Aversion that, across all users, answers for Automation Bias in Scenario 3 show a higher upper quartile and maximum value than Scenario 1 and 2, there is also a noticeable difference between the upper quartiles for Scenarios 2 and 3 for detrimental algorithmic aversion (See Figure8.6).

This suggests that as the user knows less of their environment, there is a slight increase in the amount of mistakes due to some users over relying on the system during Scenario 3 while at the same time, mistakes made from refusing to rely on the system in Scenarios 2 and 3 also

8. *Measuring User Reliance and Technology Dominance in Waterborne Deep VPR With Human-Centered AI*

see a subtle increase.

**RQ2: How is Technology Dominance affected by Human-Centered AI methods?:**

Human-Centered AI methods	Value	Num DF	Den DF	F Value	Pr > F
Wilks' lambda	0.6988	2.0000	148.0000	31.8949	0.0000
Pillai's trace	0.3012	2.0000	148.0000	31.8949	0.0000

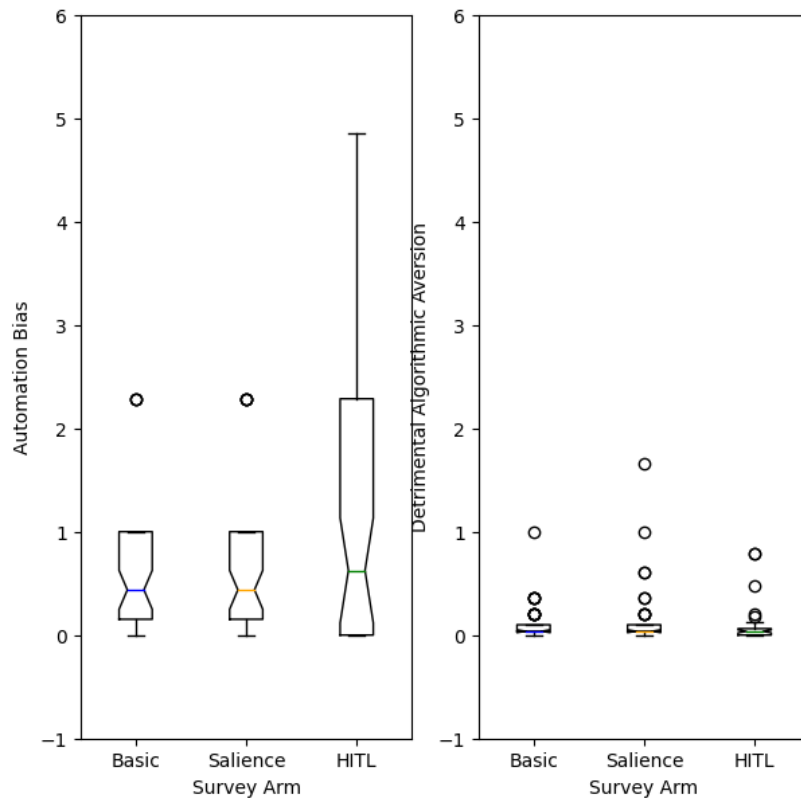


Figure 8.7: Left: Box Plots for Automation Bias ratio for all users across Human-Centered AI methods arms Right: Box Plots for Detrimental Algorithmic Aversion ratio for all users across Human-Centered AI methods arms.

Results for Technology Dominance over different Human-Centered AI methods also show significant difference, but with a Wilk's lambda value of 0.6988 and a Pillai's trace of 0.3012. Once again, the box plots for Automation Bias and Detrimental Algorithmic Aversion across users answers within the three study arms relating to Human-Centered AI methods are similar, with a potentially higher Automation Bias in Human-in-the-loop interactivity which shows a higher upper quartile and maximum value (See Figure8.7). This suggests that HITL systems can increase decision mistakes made by users relying on the AI system.

8. *Measuring User Reliance and Technology Dominance in Waterborne Deep VPR With Human-Centered AI*

Moving over to user reliance, I perform a similar quantitative analysis based on the results of my online survey by performing Multivariate analysis of variance (MANOVA) where the dependent variables are the six trust measurement likert scale values across the three independent variables: *Scenario*, *Retrieval Quality* and *Human-Centered AI methods*.

**RQ3: How is User Reliance affected by Scenario?:**

Scenario	Value	Num DF	Den DF	F Value	Pr > F
Wilks' lambda	0.1675	6.0000	2710.0000	2244.5019	0.0000
Pillai's trace	0.8325	6.0000	2710.0000	2244.5019	0.0000

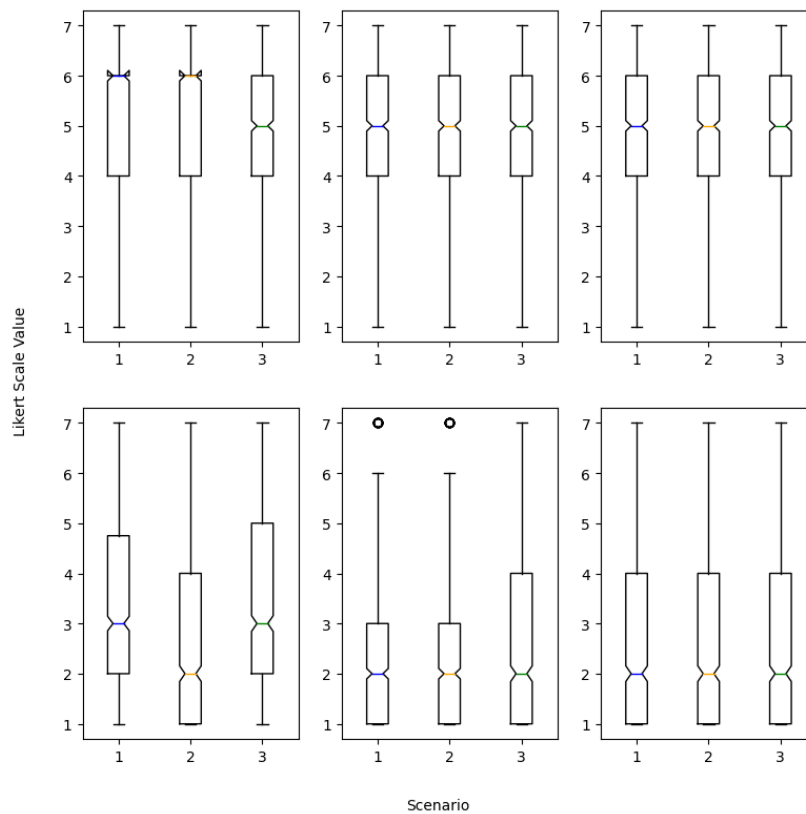


Figure 8.8: Top Row: Box Plots for Positive User Reliance prompts (Fields 3-5) across Scenarios 1-3  
 Right: Box Plots for Negative User Reliance prompts (Fields 6-8) across Scenarios 1-3.

Results for user reliance appear to be statistically significant, Wilks' lambda stands at 0.1675 while Pillai's trace is 0.8325, which further enhances the evidence for this. Despite this, the box plots are difficult to analyse in terms of change as the top row of plots show user responses to fields associated with positive reliance are near identical, plots on the bottom row

8. *Measuring User Reliance and Technology Dominance in Waterborne Deep VPR With Human-Centered AI*

however so show differences with regards to negative reliance fields with Scenarios 1 and 3 having higher values for Field 6: “I am suspicious of the system’s output” and Field 7: “The system is deceptive” (See Figure8.8).

This suggests that XAI was able to decrease users suspicion towards the system somewhat although it did not change the perceived “deceitfulness” of the AI, whereas HITL unfortunately did slightly increase this latter feeling across the users.

**RQ4: How is User Reliance affected by Retrieval Quality?:**

Scenario	Value	Num DF	Den DF	F Value	Pr > F
Wilks’ lambda	0.1550	6.0000	2710.0000	2462.0118	0.0000
Pillai’s trace	0.8450	6.0000	2710.0000	2462.0118	0.0000

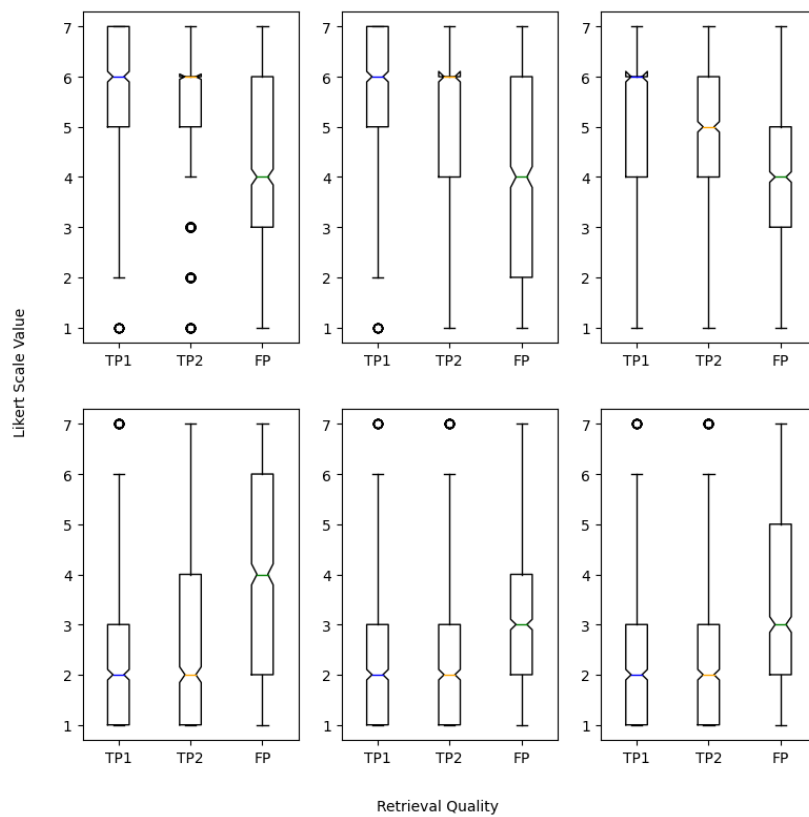


Figure 8.9: Top Row: Box Plots for Positive User Reliance prompts (Fields 3-5) across Retrieval Quality 1-3 Right: Box Plots for Negative User Reliance prompts (Fields 6-8) across Retrieval Quality 1-3.

For RQ4, results are similar to that of RQ3, Wilks’ lambda 0.1550 and Pillai’s trace of 0.8450 suggest strong evidence along with a P value of 0 indicating statistical significance. The

8. *Measuring User Reliance and Technology Dominance in Waterborne Deep VPR With Human-Centered AI*

---

box plots for this variable appear much more clear; Retrieval Quality 3 (i.e. False Positives) consistently incited lower and higher means and quartiles for positive/negative reliance fields respectively. Retrieval Quality 2, Sub-optimal True Positives, show a similar but less significant trend for Fields 3, 4, 5 and 6.

This is a positive sign as we would expect users to rely upon the AI less when experiencing False Positive and lower quality True Positive outputs, which the lower values for trust, reliability and higher values for distrust, suspiciousness etc. indicate, although ideally one may hope that the differences were further apart.

**RQ5: How is User Reliance affected by Human-Centered AI methods?:**

Scenario	Value	Num DF	Den DF	F Value	Pr > F
Wilks' lambda	0.1540	6.0000	2710.0000	2481.1073	0.0000
Pillai's trace	0.8460	6.0000	2710.0000	2481.1073	0.0000

Finally, results for RQ5 follow the pattern of RQ3 and 4 with a noticeable Wilks' lambda of 0.1540 and Pillai's trace of 0.8460 while being statistically significant. Box plots for each user reliance field show patterns such as Human-in-the-loop having a higher lower quartile for Fields 3 and 4 with values 1-3 only being outlier values compared to the other two arms, for Fields 5, 6 and 7 basic and human-in-the-loop appear to share the same statistics with the latter having a lower upper quartile for Field 8.

Explainable AI on the other hand consistently has a slight negative effect on reliance in Fields 4 through 8, having lower values for the lower quartile in Fields 4 and 5, and, having higher median and upper quartiles in Fields 6, 7 and 8.

This indicates there may be a small increase/decrease in positive/negative user reliance respectively for human in the loop and XAI.

**8.3.2 Qualitative Analysis**

With the interviews transcribed, I took related quotes from my participants and used these to form a collection of 2nd order themes, from which I gained five major themes highlighted by the interviews with regards to user reliance and automation bias for waterborne navigation.

The first major theme across the interviews was that of users **Perceived System Limitations**, given only a quick introduction to Deep VPR the participants were able to spot a number of both potential and real limitations of these systems, including reliance on high visibility input images for good results and reliance on a pre-built retrieval database limiting overall scope.



8. Measuring User Reliance and Technology Dominance in Waterborne Deep VPR With Human-Centered AI

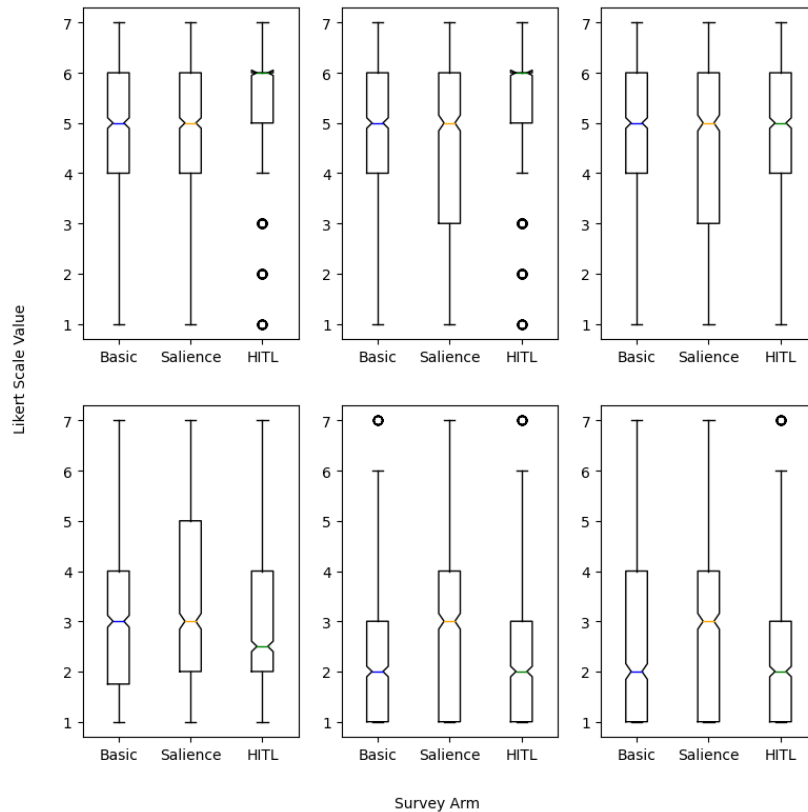


Figure 8.10: Top Row: Box Plots for Positive User Reliance prompts (Fields 3-5) across Human-Centered AI methods arms: Box Plots for Negative User Reliance prompts (Fields 6-8) across Human-Centered AI methods arms.

*Participant 1: “My biggest concerns are that it is reliant on good visibility. Which isn’t regularly something you can rely on at sea”, Participant 4: “The tool seems limited to a specific area so I can imagine issues expanding its area of operation”.*

A more domain specific issue that was commonly raised was that, due to Deep VPR only giving an approximate position based on a matching image, it would be risky to make use of these in tighter coastal areas as small differences in positioning can result in an accident. *Participant 2: “I assume it only works in coastal areas however, which also tends to need more precise fixes so I would hope to use it alongside other methods to get a good fix”, Participant 3: “I’d be trusting it more offshore than in harbour, small differences can make a big difference in tight spaces”.*

As Participant 2 alludes to in the quote above, another major theme was **Working Alongside Manual Navigation Methods**, many of the interview participants had experience in mar-

## 8. *Measuring User Reliance and Technology Dominance in Waterborne Deep VPR With Human-Centered AI*

---

itime navigation and as such were familiar with such methods.

When in a position where they lacked knowledge of their own position due to GNSS denial, these participants made the argument that because they would always have access to traditional methods, they would never need to fully rely on the AI in any scenario as, given additional manual effort, a mariner can always calculate a precise localization.

*Participant 1: “I would never rely on any piece of technology completely, including GNSS. As someone familiar with the area I could easily guess as to whether the AI was correct or not no matter the scenario”. Participant 4: “Even when you don’t know your position on the map you can see the land around you, so even in the worst examples I feel I could have used a static map to do traditional navigation as an alternative”.*

Of course, because of this, participants opinions on where AI would fit into waterborne navigation alongside traditional methods were mixed. *Participant 2: “I think it was quite good, mostly accurate as well, there are many use cases for GNSS denied navigation which this may be good”*, *Participant 4: “Harbours often have built in features and harbour lights etc. so I’m not sure where the AI fits in, it may be worth having the AI work off of these features instead”*, *Participant 5: “If it could get to a high enough standard then it could compete, but the floor for navigation accuracy is extremely high due to other technologies such as RADAR and LIDAR”.*

The third major theme was that of **Exposing underlying systems**, when asked about the effectiveness of both Explainable AI and Human-in-the-loop, users would often point out how they felt each method was able to reduce the black box nature of these models, which was generally regarded as a positive change. *Participant 2: “I think [Image Saliency] fits quite well within the field of XAI which has had more of an uptake recently and is effective at communicating the underlying workings of the model”*, *Participant 1: “(Human-in-the-loop) seemed to make the model perform better so it gave me more confidence in the model I would say, it did not make me judge it in a negative way and exposing the decision making once again was overall positive”.*

In addition, I found **Explainable AI divisive among end users**, most users agree that it effectively exposed the underlying features used to match an image but whether or not this was a positive change varied, although users unanimously agreed that what the system saw differed from what they would use for navigation or even replicating the machines task of image matching.

*Participant 5: “When I could see the AI’s decision making process, correct ones were gen-*

## 8. Measuring User Reliance and Technology Dominance in Waterborne Deep VPR With Human-Centered AI

---

erally useful, however, many of the fail cases ranged from understandable to nowhere near related and seemingly random. It seemed logical but not what I personally would have chosen”, Participant 3: “I found that more subtle shape highlights were useful for example, as it did not immediately jump out to me. It seemed that conspicuous features are different between the human and AI. But it wasn’t always correct”.

Finally, the last major theme was that **Human-in-the-loop is useful for end users**, all but one participant felt that my novel Human-in-the-loop approach, which was essentially an active learning style segmentation map editing stage, was worth using across various examples. The most common use case was for object removal, in this case boats, many users would draw a segmentation around these objects so that they would not be included and reported increased accuracy as a result.

Participant 2: “I would intervene around two thirds of the time, namely when there were boats in the image [...] my expectations of the model were higher as I am now putting in personal effort to fine tune it, generally this would be the case as the accuracy would increase after doing such”, Participant 4: “I intervened when about 40% of the “land” was clearly not, often when a boat was contained within the land highlight I would go back and change it”.

Only one participant objected to Human-in-the-loop, on the grounds that it may be difficult to engage with it practically while also navigating a vessel. Participant 3: “I wouldn’t expect a navigator onboard a vessel to be manually drawing in land segmentation on each (or indeed any) image”.

## 8.4 Conclusions

I have presented a set of results from three online surveys measuring User Reliance and Technology Dominance in the domain of Waterborne Deep VPR, along with qualitative interviews with a small group of real world mariners.

From the surveys, I find that reliance and technology dominance show statistically significant changes across scenario, retrieval quality or Human-Centered AI methods, although the changes were always quite subtle. My interview participants were able to shed more light on these findings, pointing out that mariners travelling along coastal areas should always have access to manual navigation tools and thus be able to get a precise locational fix. As such, the scenarios, especially that of having no knowledge of ones position, may not be representative of the situations mariners find themselves in.

## *8. Measuring User Reliance and Technology Dominance in Waterborne Deep VPR With Human-Centered AI*

---

As already stated in relation to retrieval quality, when presented with an equal distribution of Optimal True Positives, Suboptimal True Positives and False Positives, user reliance predictably showed a tendency to drop for the latter two categories. Interview Participants highlighted that False Positives were simple to identify without prior knowledge due to their clear visual differences, with user feelings towards the model after seeing false positives varying on a per user basis.

My main topic of interest in improving user reliance and technology dominance was the incorporation of both Explainable AI and Human-in-the-loop style interaction. The survey highlighted some statistically significant trends here, as my Explainable AI method showed similar if not slightly worsened levels of reliance compared to baseline with the Human-in-the-loop variant having an opposite effect, trending towards a slight improvement to user reliance.

The interviewees had a lot to say about these methods, agreeing that both helped to reveal the underlying AI systems in different ways, with Explainable AI displaying which pixel-wise features were used for decision making in a human interpretable way and Human-in-the-loop explicitly revealing the semantic segmentation stage, allowing the user to interact with this intermediate stage of the overall Deep VPR pipeline.

Whether this had a positive or negative impact however depends on the result, XAI was divisive among the participants who generally found the highlighting of what the AI sees to differ from what they expected, which was appreciated by some but considered jarring by others. Human-in-the-loop was generally considered a positive addition, with users feeling that it gave them agency to “fix” problems such as erroneous boat detection and thus receive a better result.

Overall, I believe that I have presented some interesting initial findings into user reliance and technology dominance for Deep VPR as navigational decision-support. We believe that scenarios more closely related to GNSS denial situations, such as spoofing, may provoke more significant changes in user reliance upon the AI. More intensive human-in-the-loop inclusions, such as dedicated active learning stages, may also reveal more about how end users feel about intervening with AI models to ensure accurate results.

## Chapter 9

# Conclusions and Future Work

### Contents

---

9.1	Conclusions . . . . .	161
9.2	Contributions . . . . .	164
9.3	Future Work . . . . .	165

---

## 9.1 Conclusions

I have presented one of the first major works on the viability of Deep Visual Place Recognition in Waterborne Domains, from this I now know that state-of-the-art Deep VPR models such as SSM-VPR, which is built on a VGG16 backbone trained on the terrestrial Places365 dataset, was able to effectively translate to the bucolic waterborne imagery of Symphony Lake and by applying explainable image saliency, I was able to view the different types of features given weight by the model for this type of imagery, which just so happened to be notable objects that stand out from the visible land line.

Of course, I was not satisfied with the Symphony Lake dataset, as although it met the requirement of being on water, it was still captured within a relatively small contained lake area. To remedy this, I worked in collaboration with Marine AI, a Plymouth based software company that worked on IBM/Promare Mayflower Autonomous Ship (MAS), to build the Plymouth Sound Dataset which covers the maritime region of the same name over the course of several days. This dataset allowed us to carry out more legitimate Waterborne Deep VPR evaluation as the data is much more relevant to my original motivations and objectives; helping GNSS denied vessels in coastal regions. This dataset also acts as one of the first benchmarks datasets

for Waterborne Deep VPR, allowing potential future researchers in the field to measure the success of further developments in Deep VPR in coastal areas.

Upon re-evaluating Deep VPR performance on the Plymouth Sound Dataset, I made more discoveries regarding the challenges of such data including the natural scarcity of useful information for retrieval contained within many coastal imagery due to large amounts of uninformative sea and sky information, and as a result, less useful land information. I also found that as a vessel moves further from the shoreline, this issue is compounded as land features along the shore would become progressively smaller and far less frequent as the boundary between shoreline and open sea was crossed.

In this work I attempted to tackle this problem using two separate Computer Vision techniques; Region Proposal and Image Segmentation. For Region Proposal, my intention was to find an unsupervised algorithm that could build bounding boxes along the visible land region only, allowing for an enriched subset of local descriptors not influenced by the sky and sea features, the reason I chose not to use supervised methods such as Faster R-CNN [16] was due to them being trained on object oriented datasets, meaning they would likely not translate well to detecting generic land strips. Unfortunately, this methodology performed marginally worse than baseline SSM-VPR, as such I would not advise the use of unsupervised Region Proposal for Waterborne Deep VPR in future and I would recommend the training of a supervised model, trained to detect visible land strips.

Thankfully, Image Segmentation proved to be a positive enhancement for Waterborne Deep VPR through the use of WaSR [37] for land, sea and sky segmentation. This acts as a powerful knowledge prior able to be used in a number of different ways, three of which I have presented in this work, including segmentation enhanced feature maps to enrich the convolutional land features produced by the backbone network, semantically aware local descriptor refinement to automatically discard descriptors generated from bounding boxes not containing useful information and my novel horizon based spatial matching method, which took inspiration from both WASABI and SSM-VPR Stage 2 to create a more efficient and discriminative method for scoring waterborne image spatial similarity by only covering spatial features along the semantic land line.

From a technical standpoint then, my findings for creating Optimal Waterborne Deep VPR show that the most discriminative features for retrieval are always located along the local land line, as this is where most of the notable shapes and structures contrast against the sky background, producing discriminative convolutional features. On the flipside, sea and sky tend to be

visually homogenous and, naturally, do not provide additional information for localization, sky is also a highly variable visual feature globally so great care must be taken to ensure the Deep VPR model can train on and store examples of locations under different weather conditions. At the time of writing, Image Segmentation of land, sea and sky appears to be the strongest tool for exploiting both of these major findings, as it allows pixel-wise enhancement of land and reduction of features from sea and sky.

Knowing that AI tools intended for navigation based tasks have been historically controversial among human end users, I additionally carried out an online survey analysing user trust and technology dominance (As defined by Cabitza et al [105]) against my Waterborne Deep VPR pipeline. I believe this study to be an essential piece of the overall work, as it gives us insight into how we can ultimately get end users to engage with and adopt Deep VPR for navigational decision support.

Knowing that Explainable AI and Human-in-the-loop methods have seen success in the past for decision support systems, I created three arms covering the basecase with no Human-AI interaction, interaction through Explainable AI using my previously developed novel Image Saliency and Human-in-the-loop by allowing the participants to engage with the intermediate stage of my final pipeline which used WaSR to highlight visible land, showing them the resulting segmentation and asking if they would want to manually label for the purpose of correction or improvement. To gain insight into user trust and technology dominance across varied levels of performance and situational pressure, I generated questions with pre-determined Optimal True Positive, Sub-optimal True Positive and False Positive retrievals, and, presented each question under one of three scenarios giving users less knowledge of ground truth.

My Quantitative Analysis shows interesting patterns among the independent variables, they were deemed statistically significant through MANOVA with Wilk's lambda and Pillai's trace. The Qualitative interview participants were able to provide additional insights, which I was able to group into a set of aggregate topics. This included the user being able to quickly perceive the limitations of a Waterborne Deep VPR pipeline, including camera obstruction and need for a pre-made retrieval dataset, both of which are valid concerns for extending this Deep VPR architecture beyond the borders of the current dataset.

Other concerns included where Deep VPR as decision support would fit alongside manual navigation techniques, as participants were all well versed in maritime navigation many of them pointed out that, given some manual effort, one can always get a precise fix. Participants were still open to the idea of using Deep VPR during GNSS downtime as a defense application

however, so perhaps leaning more heavily into for Deep VPR development would be beneficial.

Finally, interviewees generally appreciated when Deep VPR pipelines reveal their underlying systems, although the difference between human salient features and features revealed to be salient by Explainable AI often differed, confusing the end user. On the other hand, using Human-in-the-loop to show users the intermediate stages and have them interact directly was seen as an overall positive, so I recommend further development of such interactions for deployment of Deep VPR as decision support in future.

## 9.2 Contributions

The main contributions can be summarized as follows:

- **Showing that State-of-the-art Deep VPR can translate from Terrestrial to Waterborne Imagery**

I compare performance and saliency of state-of-the-art place recognition on both terrestrial imagery and waterborne imagery from the Symphony Lake dataset, from this I found that state of the art was capable of translating from one domain to the other, initially this was proven using Symphony Lake although due to this image set being of a more land adjacent bucolic environment, it only revealed to us the basics of waterborne imagery such as discriminative features being located along the land line. Later evaluation on the Plymouth Sound dataset showed that pipelines could still translate, however more challenges were raised once I had moved to more coastal region imagery, which I would later find solutions for.

- **Developing the Plymouth Sound Dataset for Waterborne Deep VPR Evaluation**

Given that there was a lack of proper benchmarks for my main body of work, I created the Plymouth Sound Dataset which compared to existing waterborne image sets is made more suitable for Deep VPR evaluation due to it's focus on local shorelines over several days rather than image sets such as MaSTr1325 [53] which were developed for training object detection and collision avoidance tasks.

- **Developing Semantically Aware methods that provide state-of-the-art Waterborne Deep VPR performance**

To solve the initial challenges revealed to us upon evaluating Deep VPR on Plymouth Sound as opposed to Symphony Lake, I tried both Region Proposal and Image Segmentation techniques, finding the latter to be effective for enhancing this image domain through a number of methods which I presented in the form of my novel Deep Visual Place Recognition pipeline, Semantically Aware SSM-VPR. The pipeline minimizes redundant feature extraction



while maximizing salient feature extraction by exploiting the land segmentation mask knowledge prior in three ways; Segmentation Enhanced Feature Maps, Semantically Aware Local Descriptor Refinement and my novel method inspired by WASABI and SSM-VPR Stage 2: SHM-VPR. SHM-VPR is a particularly novel approach to using semantic segmentation, being motivated by the unique nature of waterborne imagery such that most convolutionally discriminative objects appear along the upper semantic land line edge. I evaluated Semantically Aware SSM-VPR on the Plymouth Sound dataset and compared against state of the art SSM-VPR, finding my model to outperform with regards to precision versus recall, total recall, all while reducing the impact of image segmentation computational time on inference.

- **Analysing User Trust and Technology Dominance in Waterborne Deep VPR with varying Human-AI interaction**

I presented a study to measure user trust and technology dominance in Deep VPR for navigational decision-support within a waterborne navigation context. I carried out three study arms with differing levels of Human-AI interaction, including the basecase, Explainable AI through image saliency and Human-in-the-loop interaction. To get results within a good variety of examples and situations, I assigned questions one of theoretical scenarios to measure potential changes in the users behaviour and made sure questions showed different levels of ground truth. I found that there were noticeable patterns in users responses across the independent variables, including less reliance when receiving False Positives, less suspicion brought on by XAI but also minor over-reliance when using Human-int-the-Loop. My interview stage was able to reveal a set of key insights however which can be used to influence future works in making the system ready for user adoption.

### 9.3 Future Work

Waterborne Deep VPR has the potential to be a much larger area of research than it is currently. Firstly, if we are to develop stable Waterborne Deep VPR pipelines in future, the Plymouth Sound Dataset must be expanded to incorporate vessel traversals from other international locations, right now the dataset is highly localised whereas in future we would like ships to be able to use this system while entering many ports worldwide. Of course, storage and vector search space would increase drastically upon expanding the dataset, for vector search space I propose looking more into a naval concept known as “dead reckoning” where a vessel can extrapolate an approximate area from it’s last known position and distance travelled. Knowing this area, the vector search space could be filtered to only include potential candidates whose locations

are also within the area, reducing the search space and making the retrieval of a matching candidate simpler.

For optimizing performance, I have shown that both previously developed and novel image segmentation enhancement methods used for Deep VPR are effective for waterborne given the desired labellings, however there are still challenges relating to the domain that are yet to be tackled; namely, bad weather affecting the cameras view of nearby shoreline is a common issue in the dataset, something that may be solved by image dehazing algorithms [187] which can help land line features retain contrast during periods of fog and mist.

Another challenge I do not tackle in this paper is dealing with boats, these are the most common distractor objects within waterborne imagery, taking model attention away from relevant land features. The simplest solution would be to apply an additional boat detection stage to the pipeline, either blanking these out or filling them in using image inpainting techniques [188].

Future work should definitely be considered on Human-AI interaction with Deep VPR, I did prove statistically significant findings with my online quantitative survey, but an increased sample size could help to make these results even more clear cut, although the scenario aspect did not have any significant effect. For instance, interviewees did not engage much with my scenarios as they're access to manual navigation would mean they would never feel pressured to rely on the AI completely in any circumstance, one way I could improve upon this is by making the scenarios more related to the specific GNSS denial use case, such as presenting a known path of GNSS coordinates followed by a loss of GNSS and a subsequent path based on Deep VPR retrievals.

Interviewees were also split on how the introduction of False Positives affected their opinion of the AI, some took each question as a separate instance when responding but others had their opinion lowered throughout, it may be better in future to present them with the AI working at it's true standard; perhaps by distributing True and False Positives based on the models Precision and Recall statistics.

Finally, active learning could be a great benefit for both the technological side and the Human-AI interaction side of this work, we know that knowledge priors are helpful to the model when trying to extract salient land information, so having users gradually label and re-train the dataset based on bounding boxes centered on land would be a more effective method of implementing unsupervised region proposal.

Overall there are still a vast amount of open problems to explore and tackle in Waterborne Deep VPR. I hope that by presenting one of the first major works on this specific subgroup

## *9. Conclusions and Future Work*

---

of Deep VPR, I can encourage other researchers to expand on the domain under the lens of Computer Vision. I also hope that researchers will expand upon the idea of Deep VPR user studies, as in order to make use of such systems I believe it is paramount to build a community of researchers dedicated to improving the user experience to the point where adoption becomes a real prospect.

# Bibliography

- [1] T. Naseer, G. L. Oliveira, T. Brox, and W. Burgard, “Semantics-aware visual localization under challenging perceptual conditions,” in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 2614–2620.
- [2] Y. Hou, H. Zhang, S. Zhou, and H. Zou, “Use of roadway scene semantic information and geometry-preserving landmark pairs to improve visual place recognition in changing environments,” *IEEE Access*, vol. 5, pp. 7702–7713, 2017.
- [3] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International journal of computer vision*, vol. 60, pp. 91–110, 2004.
- [4] H. Bay, T. Tuytelaars, and L. Van Gool, “Surf: Speeded up robust features,” in *Computer Vision—ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006. Proceedings, Part I 9*. Springer, 2006, pp. 404–417.
- [5] L. G. Camara and L. Přeučil, “Spatio-semantic convnet-based visual place recognition,” in *European Conference on Mobile Robots*, 2019, pp. 1–8.
- [6] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, “NetVLAD: CNN architecture for weakly supervised place recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 5297–5307.
- [7] Z. Chen, O. Lam, A. Jacobson, and M. Milford, “Convolutional neural network-based place recognition,” *arXiv preprint arXiv:1411.1509*, 2014.
- [8] H. Durrant-Whyte and T. Bailey, “Simultaneous localization and mapping: part i,” *IEEE robotics & automation magazine*, vol. 13, no. 2, pp. 99–110, 2006.
- [9] N. Snavely, S. M. Seitz, and R. Szeliski, “Photo tourism: exploring photo collections in 3d,” in *ACM siggraph 2006 papers*, 2006, pp. 835–846.

- [10] J.-Y. Jian, A. M. Bisantz, and C. G. Drury, “Foundations for an empirically determined scale of trust in automated systems,” *International journal of cognitive ergonomics*, vol. 4, no. 1, pp. 53–71, 2000.
- [11] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR’05)*, vol. 1. Ieee, 2005, pp. 886–893.
- [12] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” in *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001*, vol. 1. Ieee, 2001, pp. I–I.
- [13] J. R. R. Uijlings, K. E. A. Van De Sande, T. Gevers, and A. W. M. Smeulders, “Selective search for object recognition,” *International Journal of Computer Vision*, vol. 104, pp. 154–171, 2013.
- [14] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.
- [15] R. Girshick, “Fast R-CNN,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1440–1448.
- [16] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” *Advances in Neural Information Processing Systems*, vol. 28, 2015.
- [17] D. D. Bloisi, F. Previtali, A. Pennisi, D. Nardi, and M. Fiorini, “Enhancing automatic maritime surveillance systems with visual information,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 4, pp. 824–833, 2016.
- [18] J.-H. Kim, N. Kim, Y. W. Park, and C. S. Won, “Object detection and classification based on yolo-v5 with improved maritime dataset,” *Journal of Marine Science and Engineering*, vol. 10, no. 3, p. 377, 2022.
- [19] L. Itti, C. Koch, and E. Niebur, “A model of saliency-based visual attention for rapid scene analysis,” *IEEE Transactions on pattern analysis and machine intelligence*, vol. 20, no. 11, pp. 1254–1259, 1998.

- [20] G. Li and Y. Yu, “Visual saliency detection based on multiscale deep cnn features,” *IEEE transactions on image processing*, vol. 25, no. 11, pp. 5012–5024, 2016.
- [21] X. Qin, Z. Zhang, C. Huang, C. Gao, M. Dehghan, and M. Jagersand, “Basnet: Boundary-aware salient object detection,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 7479–7489.
- [22] R. C. Fong and A. Vedaldi, “Interpretable explanations of black boxes by meaningful perturbation,” in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 3429–3437.
- [23] V. Petsiuk, A. Das, and K. Saenko, “Rise: Randomized input sampling for explanation of black-box models,” *arXiv preprint arXiv:1806.07421*, 2018.
- [24] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, “Learning deep features for discriminative localization,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2921–2929.
- [25] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, “Grad-cam: Visual explanations from deep networks via gradient-based localization,” in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 618–626.
- [26] H. Wang, Z. Wang, M. Du, F. Yang, Z. Zhang, S. Ding, P. Mardziel, and X. Hu, “Scorecam: Score-weighted visual explanations for convolutional neural networks,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, 2020, pp. 24–25.
- [27] T. Albrecht, G. A. West, T. Tan, and T. Ly, “Visual maritime attention using multiple low-level features and naive bayes classification,” in *2011 International Conference on Digital Image Computing: Techniques and Applications*. IEEE, 2011, pp. 243–249.
- [28] A. Sobral, T. Bouwmans, and E.-h. ZahZah, “Double-constrained rpca based on saliency maps for foreground detection in automated maritime surveillance,” in *2015 12th IEEE international conference on advanced video and signal based surveillance (AVSS)*. IEEE, 2015, pp. 1–6.
- [29] J. Zhang and S. Sclaroff, “Saliency detection: A boolean map approach,” in *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 153–160.

- [30] B. Baesens, A. Adams, R. Pacheco-Ruiz, A.-S. Baesens, and S. V. Broucke, “Explainable deep learning to classify royal navy ships,” *Ieee Access*, 2023.
- [31] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.
- [32] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *Medical image computing and computer-assisted intervention—MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*. Springer, 2015, pp. 234–241.
- [33] J. Ji, X. Lu, M. Luo, M. Yin, Q. Miao, and X. Liu, “Parallel fully convolutional network for semantic segmentation,” *Ieee Access*, vol. 9, pp. 673–682, 2020.
- [34] J. Fu, J. Liu, Y. Li, Y. Bao, W. Yan, Z. Fang, and H. Lu, “Contextual deconvolution network for semantic segmentation,” *Pattern Recognition*, vol. 101, p. 107152, 2020.
- [35] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, “Pyramid scene parsing network,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2881–2890.
- [36] L. Ding, K. Zheng, D. Lin, Y. Chen, B. Liu, J. Li, and L. Bruzzone, “Mp-resnet: Multi-path residual network for the semantic segmentation of high-resolution polsar images,” *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2021.
- [37] B. Bovcon and M. Kristan, “WaSR—A water segmentation and refinement maritime obstacle detection network,” *IEEE Transactions on Cybernetics*, vol. 52, no. 12, pp. 12 661–12 674, 2021.
- [38] F. Xu, H. Uszkoreit, Y. Du, W. Fan, D. Zhao, and J. Zhu, “Explainable ai: A brief survey on history, research areas, approaches and challenges,” in *Natural language processing and Chinese computing: 8th cCF international conference, NLPCC 2019, dunhuang, China, October 9–14, 2019, proceedings, part II 8*. Springer, 2019, pp. 563–574.
- [39] R. Shenoj, J. Bowker, A. S. Dzielendziak, A. K. Lidtke, G. Zhu, F. Cheng, D. Argyos, I. Fang, J. Gonzalez, S. Johnson *et al.*, “Global marine technology trends 2030,” 2015.

- [40] I. MSC, “Maritime autonomous surface ships proposal for a regulatory scoping exercise,” MSC 98/20/2, Feb, Tech. Rep., 2017.
- [41] X. Zhang, C. Wang, L. Jiang, L. An, and R. Yang, “Collision-avoidance navigation systems for maritime autonomous surface ships: A state of the art survey,” *Ocean Engineering*, vol. 235, p. 109380, 2021.
- [42] X. Yan, F. Ma, J. Liu, and X. Wang, “Applying the navigation brain system to inland ferries,” in *Proceedings of the Conference on Computer and IT Applications in the Maritime Industries*, 2019, pp. 25–27.
- [43] J. Xue, Z. Chen, E. Papadimitriou, C. Wu, and P. H. A. J. M. Van Gelder, “Influence of environmental factors on human-like decision-making for intelligent ship,” *Ocean Engineering*, vol. 186, p. 106060, 2019.
- [44] J. Xue, C. Wu, Z. Chen, P. H. A. J. M. Van Gelder, and X. Yan, “Modeling human-like decision-making for inbound smart ships based on fuzzy decision trees,” *Expert Systems with Applications*, vol. 115, pp. 172–188, 2019.
- [45] EMSA, “Annual overview of marine casualties and incidents 2023,” 2023.
- [46] F. Cugurullo and R. A. Acheampong, “Fear of ai: an inquiry into the adoption of autonomous cars in spite of fear, and a theoretical framework for the study of artificial intelligence technology acceptance,” *AI & SOCIETY*, pp. 1–16, 2023.
- [47] J. Spravil, C. Hemminghaus, M. von Rechenberg, E. Padilla, and J. Bauer, “Detecting maritime gps spoofing attacks based on nmea sentence integrity monitoring,” *Journal of Marine Science and Engineering*, vol. 11, no. 5, p. 928, 2023.
- [48] J. S. Warner and R. G. Johnston, “Gps spoofing countermeasures,” *Homeland Security Journal*, vol. 25, no. 2, pp. 19–27, 2003.
- [49] K. Radoš, M. Brkić, and D. Begušić, “Recent advances on jamming and spoofing detection in gnss,” *Sensors*, vol. 24, no. 13, p. 4210, 2024.
- [50] K. Lazanyi and G. Maraczi, “Dispositional trust—do we trust autonomous cars?” in *2017 IEEE 15th International Symposium on Intelligent Systems and Informatics (SISY)*. IEEE, 2017, pp. 000 135–000 140.



- [51] M. Chaal, X. Ren, A. BahooToroody, S. Basnet, V. Bolbot, O. A. V. Banda, and P. Van Gelder, “Research on risk, safety, and reliability of autonomous ships: A bibliometric review,” *Safety science*, vol. 167, p. 106256, 2023.
- [52] V. Bolbot, A. Sandru, T. Saarniniemi, O. Puolakka, P. Kujala, and O. A. Valdez Banda, “Small unmanned surface vessels—a review and critical analysis of relations to safety and safety assurance of larger autonomous ships,” *Journal of Marine Science and Engineering*, vol. 11, no. 12, p. 2387, 2023.
- [53] B. Bovcon, J. Muhovič, J. Perš, and M. Kristan, “The MaSTr1325 dataset for training deep USV obstacle detection models,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2019, pp. 3431–3438.
- [54] B. Bovcon, R. Mandeljc, J. Perš, and M. Kristan, “Stereo obstacle detection for unmanned surface vehicles by imu-assisted semantic segmentation,” *Robotics and Autonomous Systems*, 2018.
- [55] Z. Chen, F. Maffra, I. Sa, and M. Chli, “Only look once, mining distinctive landmarks from convnet for visual place recognition,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2017, pp. 9–16.
- [56] D. Olid, J. M. Fácil, and J. Civera, “Single-view place recognition under seasonal changes,” *arXiv preprint arXiv:1808.06516*, 2018.
- [57] A. Khaliq, S. Ehsan, Z. Chen, M. Milford, and K. McDonald-Maier, “A holistic visual place recognition approach using lightweight CNNs for significant viewpoint and appearance changes,” *IEEE Transactions on Robotics*, vol. 36, no. 2, pp. 561–569, 2019.
- [58] S. Griffith, G. Chahine, and C. Pradalier, “Symphony lake dataset,” *International Journal of Robotics Research*, vol. 36, no. 11, pp. 1151–1158, 2017.
- [59] J. Li, R. M. Eustice, and M. Johnson-Roberson, “High-level visual features for underwater place recognition,” in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 3652–3659.
- [60] L. Thomas, M. Edwards, A. Capsey, A. Rahat, and M. Roach, “Deep visual place recognition for waterborne domains,” in *IEEE International Conference on Image Processing*, 2022, pp. 3546–3550.

- [61] L. Thomas., M. Roach., A. Rahat., A. Capsey., and M. Edwards., “Semantic and horizon-based feature matching for optimal deep visual place recognition in waterborne domains,” in *Proceedings of the 13th International Conference on Pattern Recognition Applications and Methods - ICPRAM, INSTICC*. SciTePress, 2024, pp. 761–770.
- [62] S. Lowry, N. Sünderhauf, P. Newman, J. J. Leonard, D. Cox, P. Corke, and M. J. Milford, “Visual place recognition: A survey,” *IEEE transactions on robotics*, vol. 32, no. 1, pp. 1–19, 2015.
- [63] K. Lynch, *Reconsidering the image of the city*. Springer, 1984.
- [64] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” *Advances in neural information processing systems*, vol. 25, 2012.
- [65] D. G. Lowe, “Object recognition from local scale-invariant features,” in *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2, 1999, pp. 1150–1157.
- [66] M. Brown and D. G. Lowe, “Invariant features from interest point groups.” in *Bmvc*, vol. 4, 2002, pp. 398–410.
- [67] N. Sünderhauf, S. Shirazi, A. Jacobson, F. Dayoub, E. Pepperell, B. Upcroft, and M. Milford, “Place recognition with convnet landmarks: Viewpoint-robust, condition-robust, training-free,” *Robotics: Science and Systems*, pp. 1–10, 2015.
- [68] C. L. Zitnick and P. Dollár, “Edge boxes: Locating object proposals from edges,” in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part V 13*. Springer, 2014, pp. 391–405.
- [69] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.
- [70] Sivic and Zisserman, “Video google: A text retrieval approach to object matching in videos,” in *Proceedings ninth IEEE international conference on computer vision*. IEEE, 2003, pp. 1470–1477.
- [71] D. Nister and H. Stewenius, “Scalable recognition with a vocabulary tree,” in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’06)*, vol. 2. Ieee, 2006, pp. 2161–2168.

- [72] Y. Hou, H. Zhang, and S. Zhou, "Evaluation of object proposals and convnet features for landmark-based visual place recognition," *Journal of Intelligent & Robotic Systems*, vol. 92, pp. 505–520, 2018.
- [73] H. Azizpour, A. Sharif Razavian, J. Sullivan, A. Maki, and S. Carlsson, "From generic to specific deep representations for visual recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2015, pp. 36–45.
- [74] G. Toulas, R. Slicre, and H. Jégou, "Particular object retrieval with integral max-pooling of CNN activations," *arXiv preprint arXiv:1511.05879*, 2015.
- [75] S. Ali, J. Sullivan, A. Maki, and S. Carlsson, "A baseline for visual instance retrieval with deep convolutional networks," in *Proceedings of International Conference on Learning Representations*, 2015.
- [76] A. Babenko and V. Lempitsky, "Aggregating local deep features for image retrieval," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1269–1277.
- [77] Y. Hou, H. Zhang, and S. Zhou, "Convolutional neural network-based image representation for visual loop closure detection," in *2015 IEEE international conference on information and automation*. IEEE, 2015, pp. 2238–2245.
- [78] J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Communications of the ACM*, vol. 18, no. 9, pp. 509–517, 1975.
- [79] L. Van Der Maaten, E. Postma, J. Van den Herik *et al.*, "Dimensionality reduction: a comparative," *J Mach Learn Res*, vol. 10, no. 66-71, 2009.
- [80] P. Indyk and R. Motwani, "Approximate nearest neighbors: towards removing the curse of dimensionality," in *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, 1998, pp. 604–613.
- [81] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics and Intelligent Laboratory Systems*, vol. 2, no. 1-3, pp. 37–52, 1987.
- [82] C. Masone and B. Caputo, "A survey on deep visual place recognition," *IEEE Access*, vol. 9, pp. 19 516–19 547, 2021.

- [83] E. Johns and G.-Z. Yang, “From images to scenes: Compressing an image cluster into a single scene model for place recognition,” in *2011 International conference on computer vision*. IEEE, 2011, pp. 874–881.
- [84] S. Garg, N. Suenderhauf, and M. Milford, “Lost? appearance-invariant place recognition for opposite viewpoints using visual semantics,” *arXiv preprint arXiv:1804.05526*, 2018.
- [85] O. Chum, J. Matas, and J. Kittler, “Locally optimized ransac,” in *Pattern Recognition: 25th DAGM Symposium, Magdeburg, Germany, September 10-12, 2003. Proceedings 25*. Springer, 2003, pp. 236–243.
- [86] S. Yokoo, K. Ozaki, E. Simo-Serra, and S. Iizuka, “Two-stage discriminative re-ranking for large-scale landmark retrieval,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 1012–1013.
- [87] A. Boiarov and E. Tyantov, “Large scale landmark recognition via deep metric learning,” in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2019, pp. 169–178.
- [88] L. G. Camara, C. Gäbert, and L. Přeučil, “Highly robust visual place recognition through spatial matching of CNN features,” in *IEEE International Conference on Robotics and Automation*, 2020, pp. 3748–3755.
- [89] A. Torii, R. Arandjelovic, J. Sivic, M. Okutomi, and T. Pajdla, “24/7 place recognition by view synthesis,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1808–1817.
- [90] S. Hausler, S. Garg, M. Xu, M. Milford, and T. Fischer, “Patch-netvlad: Multi-scale fusion of locally-global descriptors for place recognition,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 14 141–14 152.
- [91] H. Noh, A. Araujo, J. Sim, T. Weyand, and B. Han, “Large-scale image retrieval with attentive deep local features,” in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 3456–3465.
- [92] H. C. Longuet-Higgins, “A computer algorithm for reconstructing a scene from two projections,” *Nature*, vol. 293, no. 5828, pp. 133–135, 1981.

- [93] O. Özyeşil, V. Voroninski, R. Basri, and A. Singer, “A survey of structure from motion\*.” *Acta Numerica*, vol. 26, pp. 305–364, 2017.
- [94] M. J. Westoby, J. Brasington, N. F. Glasser, M. J. Hambrey, and J. M. Reynolds, “‘structure-from-motion’ photogrammetry: A low-cost, effective tool for geoscience applications,” *Geomorphology*, vol. 179, pp. 300–314, 2012.
- [95] G. Guidi, J.-A. Beraldin, and C. Atzeni, “High-accuracy 3d modeling of cultural heritage: the digitizing of donatello’s” maddalena,” *IEEE Transactions on image processing*, vol. 13, no. 3, pp. 370–380, 2004.
- [96] D. J. Yeong, G. Velasco-Hernandez, J. Barry, and J. Walsh, “Sensor and sensor fusion technology in autonomous vehicles: A review,” *Sensors*, vol. 21, no. 6, p. 2140, 2021.
- [97] Z. Wang, Y. Wu, and Q. Niu, “Multi-sensor fusion in automated driving: A survey,” *Ieee Access*, vol. 8, pp. 2847–2868, 2019.
- [98] R. E. Kalman, “A new approach to linear filtering and prediction problems,” *Transactions of the ASME—Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.
- [99] Y. Li, D. K. Jha, A. Ray, and T. A. Wettergren, “Feature level sensor fusion for target detection in dynamic environments,” in *2015 American Control Conference (ACC)*. IEEE, 2015, pp. 2433–2438.
- [100] S. Ingle and M. Phute, “Tesla autopilot: semi autonomous driving, an uptick for future autonomy,” *International Research Journal of Engineering and Technology*, vol. 3, no. 9, pp. 369–372, 2016.
- [101] P. Sun, H. Kretschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine *et al.*, “Scalability in perception for autonomous driving: Waymo open dataset,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 2446–2454.
- [102] F. Nobre, M. Kasper, and C. Heckman, “Drift-correcting self-calibration for visual-inertial slam,” in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 6525–6532.
- [103] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, “Past, present, and future of simultaneous localization and mapping: Toward

- the robust-perception age,” *IEEE Transactions on robotics*, vol. 32, no. 6, pp. 1309–1332, 2016.
- [104] T. Araujo, N. Helberger, S. Kruikemeier, and C. H. De Vreese, “In ai we trust? perceptions about automated decision-making by artificial intelligence,” *AI & society*, vol. 35, no. 3, pp. 611–623, 2020.
- [105] F. Cabitza, A. Campagner, R. Angius, C. Natali, and C. Reverberi, “Ai shall have no dominion: on how to measure technology dominance in ai-supported human decision-making,” in *Proceedings of the 2023 CHI conference on human factors in computing systems*, 2023, pp. 1–20.
- [106] L. Di Lillo, T. Gode, X. Zhou, M. Atzei, R. Chen, and T. Victor, “Comparative safety performance of autonomous-and human drivers: A real-world case study of the waymo driver,” *Heliyon*, vol. 10, no. 14, 2024.
- [107] S. J. I. Yeo and W. Lin, “Autonomous vehicles, human agency and the potential of urban life,” *Geography Compass*, vol. 14, no. 10, p. e12531, 2020.
- [108] R. Hussain and S. Zeadally, “Autonomous cars: Research results, issues, and future challenges,” *IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1275–1313, 2018.
- [109] S. K. Prakash, “Artificial intelligence and autonomous vehicles with case laws,” *Indian JL & Legal Rsch.*, vol. 2, p. 1, 2021.
- [110] F. Kröger, “History of the research on vehicle automation in europe,” in *From Automated to Autonomous Driving: A Transnational Research History on Pioneers, Artifacts and Technological Change (1950-2000)*. Springer, 2024, pp. 165–253.
- [111] T. Sheridan, “Trustworthiness of command and control systems,” in *Analysis, Design and Evaluation of Man–Machine Systems 1988*. Elsevier, 1989, pp. 427–431.
- [112] F. Bousetouane and B. Morris, “Fast cnn surveillance pipeline for fine-grained vessel classification and detection in maritime scenarios,” in *2016 13th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. IEEE, 2016, pp. 242–248.

- [113] T. Cane and J. Ferryman, “Saliency-based detection for maritime object tracking,” in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2016, pp. 18–25.
- [114] M. Kristan, V. S. Kenk, S. Kovačič, and J. Perš, “Fast image-based obstacle detection from unmanned surface vehicles,” *IEEE Transactions on Cybernetics*, vol. 46, no. 3, pp. 641–654, 2015.
- [115] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld, “Face recognition: A literature survey,” *ACM computing surveys (CSUR)*, vol. 35, no. 4, pp. 399–458, 2003.
- [116] I. Adjabi, A. Ouahabi, A. Benzaoui, and A. Taleb-Ahmed, “Past, present, and future of face recognition: A review,” *Electronics*, vol. 9, no. 8, p. 1188, 2020.
- [117] P. Dollar, C. Wojek, B. Schiele, and P. Perona, “Pedestrian detection: An evaluation of the state of the art,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 4, pp. 743–761, 2011.
- [118] A. A. Alsanabani, M. A. Ahmed, and A. M. Al Smadi, “Vehicle counting using detecting-tracking combinations: A comparative analysis,” in *Proceedings of the 2020 4th International Conference on Video and Image Processing*, 2020, pp. 48–54.
- [119] Z. Zou, K. Chen, Z. Shi, Y. Guo, and J. Ye, “Object detection in 20 years: A survey,” *Proceedings of the IEEE*, vol. 111, no. 3, pp. 257–276, 2023.
- [120] Y. Freund and R. E. Schapire, “A decision-theoretic generalization of on-line learning and an application to boosting,” *Journal of computer and system sciences*, vol. 55, no. 1, pp. 119–139, 1997.
- [121] P. F. Felzenszwalb and D. P. Huttenlocher, “Efficient graph-based image segmentation,” *International journal of computer vision*, vol. 59, pp. 167–181, 2004.
- [122] S. Manen, M. Guillaumin, and L. Van Gool, “Prime object proposals with randomized prim’s algorithm,” in *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 2536–2543.
- [123] P. Dollár and C. L. Zitnick, “Structured forests for fast edge detection,” in *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 1841–1848.

- [124] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.
- [125] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, “Backpropagation applied to handwritten zip code recognition,” *Neural computation*, vol. 1, no. 4, pp. 541–551, 1989.
- [126] M. Everingham, “The PASCAL visual object classes challenge 2008 (VOC2008) results,” in <http://www.pascal-network.org/challenges/VOC/voc2008/year=workshop/index.html>, 2008.
- [127] H. V. Vo, P. Pérez, and J. Ponce, “Toward unsupervised, multi-object discovery in large-scale image collections,” in *Proceedings of the European Conference on Computer Vision*, 2020, pp. 779–795.
- [128] F. Chazal, L. J. Guibas, S. Y. Oudot, and P. Skraba, “Persistence-based clustering in Riemannian manifolds,” *Journal of the ACM*, vol. 60, no. 6, pp. 1–38, 2013.
- [129] D. K. Prasad, H. Dong, D. Rajan, and C. Quek, “Are object detection assessment criteria ready for maritime computer vision?” *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 12, pp. 5295–5304, 2019.
- [130] 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,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 8, pp. 1993–2016, 2017.
- [131] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, “mixup: Beyond empirical risk minimization,” *arXiv preprint arXiv:1710.09412*, 2017.
- [132] I. Ullah, M. Jian, S. Hussain, J. Guo, H. Yu, X. Wang, and Y. Yin, “A brief survey of visual saliency detection,” *Multimedia Tools and Applications*, vol. 79, pp. 34 605–34 645, 2020.
- [133] T. N. Mundhenk, B. Y. Chen, and G. Friedland, “Efficient saliency maps for explainable ai,” *arXiv preprint arXiv:1911.11293*, 2019.



- [134] W. Schneider and R. M. Shiffrin, "Controlled and automatic human information processing: I. detection, search, and attention." *Psychological review*, vol. 84, no. 1, p. 1, 1977.
- [135] M.-M. Cheng, N. J. Mitra, X. Huang, P. H. Torr, and S.-M. Hu, "Global contrast based salient region detection," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 3, pp. 569–582, 2014.
- [136] M. Runxin, Y. Yang, and Y. Xiaomin, "Survey on image saliency detection methods," in *2015 international conference on cyber-enabled distributed computing and knowledge discovery*. IEEE, 2015, pp. 329–338.
- [137] M. Tian, S. Luo, Y. Huang, and J. Zhao, "Extracting bottom-up attention information based on local complexity and early visual features," *Journal of computer research and development*, vol. 45, no. 10, pp. 1739–1746, 2008.
- [138] N. Sang, Z.-l. Li, and T.-x. Zhang, "Applications of human visual attention mechanisms in object detection," *Infrared and Laser Engineering*, vol. 33, no. 1, pp. 38–42, 2004.
- [139] X. Hou and L. Zhang, "Saliency detection: A spectral residual approach," in *2007 IEEE Conference on computer vision and pattern recognition*. Ieee, 2007, pp. 1–8.
- [140] S. Engel, X. Zhang, and B. Wandell, "Colour tuning in human visual cortex measured with functional magnetic resonance imaging," *Nature*, vol. 388, no. 6637, pp. 68–71, 1997.
- [141] T. Liu, Z. Yuan, J. Sun, J. Wang, N. Zheng, X. Tang, and H.-Y. Shum, "Learning to detect a salient object," *IEEE Transactions on Pattern analysis and machine intelligence*, vol. 33, no. 2, pp. 353–367, 2010.
- [142] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," *arXiv preprint arXiv:1312.6034*, 2013.
- [143] W. Xiong, Z. Xiong, and Y. Cui, "An explainable attention network for fine-grained ship classification using remote-sensing images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–14, 2022.

- [144] F. Schroff, A. Criminisi, and A. Zisserman, “Object class segmentation using random forests.” in *BMVC*, 2008, pp. 1–10.
- [145] F. Han and S.-C. Zhu, “Bottom-up/top-down image parsing with attribute grammar,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 31, no. 1, pp. 59–73, 2008.
- [146] S. Hao, Y. Zhou, and Y. Guo, “A brief survey on semantic segmentation with deep learning,” *Neurocomputing*, vol. 406, pp. 302–321, 2020.
- [147] 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 preprint arXiv:1704.06857*, 2017.
- [148] H. Noh, S. Hong, and B. Han, “Learning deconvolution network for semantic segmentation,” in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1520–1528.
- [149] V. Badrinarayanan, A. Kendall, and R. Cipolla, “Segnet: A deep convolutional encoder-decoder architecture for image segmentation,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.
- [150] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, “Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 834–848, 2017.
- [151] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [152] F. Yu and V. Koltun, “Multi-scale context aggregation by dilated convolutions,” *arXiv preprint arXiv:1511.07122*, 2015.
- [153] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello, “Enet: A deep neural network architecture for real-time semantic segmentation,” *arXiv preprint arXiv:1606.02147*, 2016.

- [154] S. Mehta, M. Rastegari, A. Caspi, L. Shapiro, and H. Hajishirzi, “Espnet: Efficient spatial pyramid of dilated convolutions for semantic segmentation,” in *Proceedings of the european conference on computer vision (ECCV)*, 2018, pp. 552–568.
- [155] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, “Scene parsing through ade20k dataset,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 633–641.
- [156] B. Bovcon, J. Perš, M. Kristan *et al.*, “Stereo obstacle detection for unmanned surface vehicles by IMU-assisted semantic segmentation,” *Robotics and Autonomous Systems*, vol. 104, pp. 1–13, 2018.
- [157] K. He, X. Zhang, S. Ren, and J. Sun, “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification,” in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.
- [158] B. Shneiderman, *Human-centered AI*. Oxford University Press, 2022.
- [159] T. Capel and M. Brereton, “What is human-centered about human-centered ai? a map of the research landscape,” in *Proceedings of the 2023 CHI conference on human factors in computing systems*, 2023, pp. 1–23.
- [160] D. Castelvechi, “Can we open the black box of ai?” *Nature News*, vol. 538, no. 7623, p. 20, 2016.
- [161] E. Mosqueira-Rey, E. Hernández-Pereira, D. Alonso-Ríos, J. Bobes-Bascarán, and Á. Fernández-Leal, “Human-in-the-loop machine learning: a state of the art,” *Artificial Intelligence Review*, vol. 56, no. 4, pp. 3005–3054, 2023.
- [162] F. M. Zanzotto, “Human-in-the-loop artificial intelligence,” *Journal of Artificial Intelligence Research*, vol. 64, pp. 243–252, 2019.
- [163] D. Gunning and D. Aha, “Darpa’s explainable artificial intelligence (xai) program,” *AI magazine*, vol. 40, no. 2, pp. 44–58, 2019.
- [164] L. A. Hendricks, Z. Akata, M. Rohrbach, J. Donahue, B. Schiele, and T. Darrell, “Generating visual explanations,” in *Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*. Springer, 2016, pp. 3–19.

- [165] X. Wu, L. Xiao, Y. Sun, J. Zhang, T. Ma, and L. He, “A survey of human-in-the-loop for machine learning,” *Future Generation Computer Systems*, vol. 135, pp. 364–381, 2022.
- [166] S. Budd, E. C. Robinson, and B. Kainz, “A survey on active learning and human-in-the-loop deep learning for medical image analysis,” *Medical image analysis*, vol. 71, p. 102062, 2021.
- [167] C. Chai and G. Li, “Human-in-the-loop techniques in machine learning,” *IEEE Data Eng. Bull.*, vol. 43, no. 3, pp. 37–52, 2020.
- [168] R. M. Monarch, *Human-in-the-Loop Machine Learning: Active learning and annotation for human-centered AI*. Simon and Schuster, 2021.
- [169] B. Settles, “From theories to queries: Active learning in practice,” in *Active learning and experimental design workshop in conjunction with AISTATS 2010*. JMLR Workshop and Conference Proceedings, 2011, pp. 1–18.
- [170] L. Yang, Y. Zhang, J. Chen, S. Zhang, and D. Z. Chen, “Suggestive annotation: A deep active learning framework for biomedical image segmentation,” in *Medical Image Computing and Computer Assisted Intervention- MICCAI 2017: 20th International Conference, Quebec City, QC, Canada, September 11-13, 2017, Proceedings, Part III 20*. Springer, 2017, pp. 399–407.
- [171] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, “Vision meets robotics: The KITTI dataset,” *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.
- [172] N. Sünderhauf, P. Neubert, and P. Protzel, “Are we there yet? challenging seqslam on a 3000 km journey across all four seasons,” in *Proc. of workshop on long-term autonomy, IEEE international conference on robotics and automation (ICRA)*. Citeseer, 2013, p. 2013.
- [173] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, “Places: A 10 million image database for scene recognition,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 6, pp. 1452–1464, 2017.
- [174] K. A. Tsintotas, L. Bampis, and A. Gasteratos, “The revisiting problem in simultaneous localization and mapping: A survey on visual loop closure detection,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 11, pp. 19 929–19 953, 2022.

- [175] R. B. Sousa, H. M. Sobreira, and A. P. Moreira, “A systematic literature review on long-term localization and mapping for mobile robots,” *Journal of Field Robotics*, vol. 40, no. 5, pp. 1245–1322, 2023.
- [176] A. Benbihi, S. Arravechia, M. Geist, and C. Pradalier, “Image-based place recognition on bucolic environment across seasons from semantic edge description,” in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 3032–3038.
- [177] Y. He, X. Zhang, and J. Sun, “Channel pruning for accelerating very deep neural networks,” in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 1389–1397.
- [178] A. López-Cifuentes, M. Escudero-Vinolo, J. Bescós, and Á. García-Martín, “Semantic-aware scene recognition,” *Pattern Recognition*, vol. 102, p. 107256, 2020.
- [179] P. Panphattarasap and A. Calway, “Visual place recognition using landmark distribution descriptors,” in *Computer Vision—ACCV 2016: 13th Asian Conference on Computer Vision, Taipei, Taiwan, November 20–24, 2016, Revised Selected Papers, Part IV 13*. Springer, 2017, pp. 487–502.
- [180] D. P. Robertson and R. Cipolla, “An image-based system for urban navigation.” in *Bmvc*, vol. 19, no. 51, 2004, p. 165.
- [181] S. Koul and A. Eydgahi, “The impact of social influence, technophobia, and perceived safety on autonomous vehicle technology adoption,” *Periodica Polytechnica Transportation Engineering*, vol. 48, no. 2, pp. 133–142, 2020.
- [182] D. Kahneman, O. Sibony, and C. R. Sunstein, *Noise: A flaw in human judgment*. Hachette UK, 2021.
- [183] M. S. Silberman, B. Tomlinson, R. LaPlante, J. Ross, L. Irani, and A. Zaldivar, “Responsible research with crowds: pay crowdworkers at least minimum wage,” *Communications of the ACM*, vol. 61, no. 3, pp. 39–41, 2018.
- [184] A. Smith, V. Kumar, J. Boyd-Graber, K. Seppi, and L. Findlater, “Closing the loop: User-centered design and evaluation of a human-in-the-loop topic modeling system,” in *23rd International Conference on Intelligent User Interfaces*, 2018, pp. 293–304.

- [185] D. Honeycutt, M. Nourani, and E. Ragan, “Soliciting human-in-the-loop user feedback for interactive machine learning reduces user trust and impressions of model accuracy,” in *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, vol. 8, 2020, pp. 63–72.
- [186] Q. V. Liao, D. Gruen, and S. Miller, “Questioning the ai: informing design practices for explainable ai user experiences,” in *Proceedings of the 2020 CHI conference on human factors in computing systems*, 2020, pp. 1–15.
- [187] B. Li, W. Ren, D. Fu, D. Tao, D. Feng, W. Zeng, and Z. Wang, “Benchmarking single-image dehazing and beyond,” *IEEE Transactions on Image Processing*, vol. 28, no. 1, pp. 492–505, 2018.
- [188] Z. Qin, Q. Zeng, Y. Zong, and F. Xu, “Image inpainting based on deep learning: A review,” *Displays*, vol. 69, p. 102028, 2021.