Multimodal learning from visual and remotely sensed data

Dushyant Rao, BE (Hons 1) BSc, MSc

A thesis submitted in fulfillment
of the requirements of the degree of
Doctor of Philosophy

THE UNIVERSITY OF
SYDNEY

Australian Centre for Field Robotics
School of Aerospace, Mechanical and Mechatronic Engineering
The University of Sydney

July 2016

Declaration

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the University or other institute of higher learning, except where due acknowledgement has been made in the text.

Dushyant Rao, BE (Hons 1) BSc, MSc

9 July 2016

Abstract

Dushyant Rao, BE (Hons 1) BSc, MSc
The University of Sydney

Doctor of Philosophy
July 2016

Multimodal learning from visual and remotely sensed data

Autonomous vehicles are often deployed to perform exploration and monitoring missions in unseen environments. In such applications, there is often a compromise between the information richness and the acquisition cost of different sensor modalities. Visual data is usually very information-rich, but requires in-situ acquisition with the robot. In contrast, remotely sensed data has a larger range and footprint, and may be available prior to a mission. In order to effectively and efficiently explore and monitor the environment, it is critical to make use of all of the sensory information available to the robot.

One important application is the use of an Autonomous Underwater Vehicle (AUV) to survey the ocean floor. AUVs can take high resolution in-situ photographs of the sea floor, which can be used to classify different regions into various habitat classes that summarise the observed physical and biological properties. This is known as *benthic habitat mapping*. However, since AUVs can only image a tiny fraction of the ocean floor, habitat mapping is usually performed with remotely sensed bathymetry (ocean depth) data, obtained from shipborne multibeam sonar.

With the recent surge in unsupervised feature learning and deep learning techniques, a number of previous techniques have investigated the concept of *multimodal learning*: capturing the relationship between different sensor modalities in order to perform classification and other inference tasks. This thesis proposes related techniques for visual and remotely sensed data, applied to the task of autonomous exploration and

monitoring with an AUV. Doing so enables more accurate classification of the benthic environment, and also assists autonomous survey planning.

The first contribution of this thesis is to apply unsupervised feature learning techniques to marine data. The proposed techniques are used to extract features from image and bathymetric data separately, and the performance is compared to that with more traditionally used features for each sensor modality.

The second contribution is the development of a multimodal learning architecture that captures the relationship between the two modalities. The model is robust to missing modalities, which means it can extract better features for large-scale benthic habitat mapping, where only bathymetry is available. The model is used to perform classification with various combinations of modalities, demonstrating that multimodal learning provides a large performance improvement over the baseline case.

The third contribution is an extension of the standard learning architecture using a gated feature learning model, which enables the model to better capture the 'one-to-many' relationship between visual and bathymetric data. This opens up further inference capabilities, with the ability to predict visual features from bathymetric data, which allows image-based queries. Such queries are useful for AUV survey planning, especially when supervised labels are unavailable.

The final contribution is the novel derivation of a number of information-theoretic measures to aid survey planning. The proposed measures predict the utility of unobserved areas, in terms of the amount of expected additional visual information. As such, they are able to produce utility maps over a large region that can be used by the AUV to determine the most informative locations from a set of candidate missions.

The models proposed in this thesis are validated through extensive experiments on real marine data. Furthermore, the introduced techniques have applications in various other areas within robotics. As such, this thesis concludes with a discussion on the broader implications of these contributions, and the future research directions that arise as a result of this work.

Acknowledgements

This thesis would not have been possible without the help and support of an enormous number of people.

First and foremost, I am eternally grateful to my supervisors, Stefan Williams and Oscar Pizarro, for providing support when I needed it, but allowing me the freedom to pursue my interests. You provided so many opportunities (very few people are privileged enough to visit the Caribbean for research purposes), and made the daunting task of completing a PhD into an incredibly fun and rewarding experience.

To all my ACFR friends (you know who you are), thanks for keeping me sane and providing the perfect balance between academic discussions and distractions. I'll truly miss our coffee breaks, despite the fact that I'm barely a coffee person - clearly there was another more important ingredient there!

To my family, thanks for keeping me grounded in reality and reminding me there's a whole wide world out there. And in particular, to Shrutie, thanks for the love and support you have given me over the years, and for happily being the breadwinner in this family!

Contents

List of Figures

List of Tables

List of Algorithms

List of Authored Publications

D. Rao, M. De Deuge, N. Nourani-Vatani, B. Douillard, S. B. Williams, O. Pizarro. **Multimodal learning for autonomous underwater vehicles from visual and bathymetric data**, in *IEEE International Conference on Robotics and Automation, 2014, pp. 3819-25.*

D. Rao, M. De Deuge, N. Nourani-Vatani, S. B. Williams, O. Pizarro. **Multimodality learning from visual and remotely sensed data**, in *IEEE/RSJ International Conference on Intelligent Robots and Systems, Workshop on Alternative Sensing and Robot Perception, 2015.*

M. S. Bewley, N. Nourani-Vatani, D. Rao, O. Pizarro, S. B. Williams. **Hierarchical classification in AUV imagery**, in *Field and Service Robotics, 2015, pp. 3-16.*

D. Rao, A. Bender, S. B. Williams, O. Pizarro. **Multimodal information-theoretic measures for autonomous exploration**, in *IEEE International Conference on Robotics and Automation, 2016, pp. 4230-37.*

D. Rao, M. De Deuge, N. Nourani-Vatani, S. B. Williams, O. Pizarro. **Multimodal learning and inference from visual and remotely sensed data**, in *International Journal of Robotics Research, under review, 2016.*

Nomenclature

Notation

$\mathbf{a}$	Visible bias vector or matrix
$\mathbf{b}$	Hidden bias vector or matrix
$E(\cdot)$	Energy function
$F(\cdot)$	Free energy function
$\mathbf{h}$	Hidden units
i	Index to an input variable
j	Index to a hidden variable
k	Index to a mixture indicator variable or label variable
$\mathbf{W}$	Weights matrix or tensor
$\mathbf{x}$	Generic input variable or visible units
$\mathbf{x}'$	Input reconstruction
$\tilde{\mathbf{x}}$	Corrupted input variable
$\mathbf{x}_{\mathcal{B}}$	Midlayer bathymetric features
$\mathbf{x}_{\mathcal{V}}$	Midlayer visual features
$\mathbf{x}^{(n)}$	n^{th} instance from an input dataset
$\mathbf{y}$	Label variable
$\mathbf{z}$	Mixture indicator variable or gating units
Z	Partition function to normalise a distribution
$\text{sigm}(\cdot)$	Logistic sigmoid function
$\mathcal{B}$	Bathymetric data
$\mathcal{B}_0$	Depth value
$\mathcal{B}_l$	Bathymetric patch (zero-meaned)
$\mathbb{E}(\cdot)$	Expectation of a distribution
$\mathbb{E}_k(\cdot)$	Expectation under the k^{th} mixture component
$\mathbb{H}(\cdot)$	Entropy of a distribution
$\mathbb{I}(\cdot\,;\cdot)$	Mutual information between two variables
$\mathcal{V}$	Visual data

Θ	A set of model parameters

Abbreviations

AE	Autoencoder
AUV	Autonomous Underwater Vehicle
CD	Contrastive Divergence
CE	Conditional entropy
CMI	Conditional mutual information
CNN	Convolutional Neural Network
DAE	Denoising Autoencoder
DEM	Digital Elevation Map
DBN	Deep Belief Network
GBM	Gated Boltzmann Machine
LIDAR	Light Detection and Ranging
MBES	Multi-beam Echosounder
MCMC	Markov Chain Monte Carlo
MixRBM	mixture of Restricted Boltzmann Machines (RBMs)
OMP	Orthogonal Matching Pursuit
PCA	Principal Components Analysis
RBM	Restricted Boltzmann Machine
ScSPM	Sparse Coding Spatial Pyramid Matching
SGD	Stochastic Gradient Descent
SLAM	Simultaneous Localisation and Mapping
SVM	Support Vector Machine

Chapter 1

Introduction

1.1 Motivation

An important capability for many autonomous vehicles is to build a semantic understanding of their surroundings when deployed in an unseen or unfamiliar environment. Self-driving cars need to identify pedestrians, signage, and other vehicles in order to navigate an urban environment safely. Indoor service robots have to detect and classify objects of interest in order to utilise them. In such applications, it is critical to make use of all sensory information available to the robot, whether it be camera images, LIDAR scans, or other remotely sensed data.

One particular application of interest in this thesis is the use of Autonomous Underwater Vehicles (AUVs) to monitor and explore the oceans. AUVs are often deployed to take high-resolution images of the seafloor along with a plethora of other sensor measurements, such as temperature, salinity and conductivity. In addition to these in-situ measurements, there is also a wealth of remotely sensed data available, most commonly in the form of multi-beam bathymetry (ocean depth) data from ship-borne sonar. This data can be used to generate *benthic habitat maps*, which classify large regions of the sea floor into broad habitat classes based on their physical and biological constituents [98]. These habitat maps are invaluable data products to marine scientists, assisting in monitoring the distribution and health of various benthic

species [82, 99, 103]. Moreover, this semantic understanding of the benthos facilitates the long term autonomy of an AUV, allowing it to perform exploration missions in line with a high-level goal (e.g. “find and monitor kelp forests”)

Benthic habitats are primarily identified by the substrate (such as rock or sediment) and the organisms present (such as algae or coral) [46], making them relatively easy to distinguish using AUV image data [89]. However, AUVs can only traverse a very small fraction of a larger area of interest, limiting the scale to which visual habitat classification can be performed. Conversely, bathymetric data is usually available a priori over an entire site, but has a low spatial resolution, on the order of metres between adjacent soundings.

In addition to benthic habitat mapping scenarios, this compromise also exists for other autonomous agents; for aerial, marine, or ground vehicles alike. Visual data is information-rich but has to be obtained in-situ. Remotely sensed data is often comparatively information-poor but has a much larger coverage and is often easier to obtain.

By modelling the relationship between visual and remotely sensed data, it is possible to leverage the benefits of each modality. Such multimodal models can handle various queries pertaining to either one or both modalities: perform classification with greater accuracy from whatever data is available [69], or predict one modality given the other [85]. From an AUV perspective, this enables more accurate habitat mapping from remotely sensed data, and allows the capability to predict what kinds of visual features might be observed in unseen dive sites given the bathymetry. Such queries aid survey planning: the model can handle queries that are class-based (e.g. “find and monitor kelp”), image-based (e.g. “find locations that are likely to look similar to this image”) or information-based (e.g. “explore areas in which the expected visual information gain is high”).

1.2 Problem statement

This thesis investigates methods to capture the joint relationship between the visual images obtained by an AUV, and remotely sensed bathymetry data obtained from shipborne sonar. By exploiting recent developments in multimodal deep learning, it is possible to build a model that facilitates both discriminative tasks (classification) and generative tasks (sampling or modality prediction).

A key driver in the development of such a model is its flexibility. As visual information is only available over a small fraction of the seafloor, the model must be able to perform inference with only bathymetry available. Indeed, it is desirable for the model to also analyse visual images alone, if necessary. Further, there also needs to be flexibility in the type of inference task. In addition to feature extraction for classification tasks, it is also desirable for the model to perform generative tasks, such as predicting visual features from bathymetry. Such capabilities allow an AUV to reason about what it might observe in previously unseen areas, and make decisions accordingly. As such, this thesis focuses on building a one-fits-all model that can handle these different types of queries, without fine tuning to a particular task.

Another important consideration is that the AUV must utilise the high-level 'intelligence' afforded by a multimodal learning model, in order to plan future actions. By jointly reasoning about visual and remotely sensed data, the AUV can then explore the environment in such a way as to optimise the information obtained through visual observation. While algorithms for AUV trajectory planning or mission planning are beyond the scope of this thesis, it is still important to consider how the proposed models could be applied to planning tasks.

1.3 Contributions

This thesis is focused on developing a multimodal model that learns the relationship between visual images and corresponding bathymetry. The primary application is for AUVs operating in marine environments, but this work also has broader implications for the robotics community. It is anticipated that the proposed techniques will act as a building block for future work in multimodal learning for ground vehicles, aerial vehicles, and other robotic platforms.

The specific aims of this thesis are as follows:

- Perform preliminary analysis of visual and bathymetric data and propose a pipeline to perform feature learning on each modality.
- Develop a multimodal learning architecture to model the relationship between the two modalities and perform classification from either or both modalities.
- Investigate models to enable additional unsupervised tasks, such as clustering and image-based queries.
- Develop techniques that use the proposed models to predict the utility of AUV candidate surveys.

Accordingly, the main contributions are the following:

- A novel application of feature learning and deep learning techniques to visual image data and shipborne multi-beam bathymetry data. The techniques are compared with traditionally used approaches, and the features extracted by the proposed method are demonstrated to perform well in classification tasks.
- A deep architecture to perform multimodal learning from both data formats. The proposed model is based on previous work in multimodal learning, and is able to perform inference when visual data is unavailable, meaning it can perform benthic habitat mapping over large regions from just the bathymetry

alone. The results demonstrate higher classification accuracy, regardless of which modalities are actually available at classification time.

- An extension of the traditional multimodal learning architecture using a gated mixture of feature learners to capture the high-level correlations, which better equips the model to handle the one-to-many relationship between visual and bathymetric data. Additional improvements are proposed to avoid specifying the number of mixture components used, and to perform inference when only bathymetric data is available. This allows the model to predict visual features from bathymetry, which facilitates image-based queries for survey planning, a useful capability when image labels are unavailable.

- Novel derivations of a number of information-theoretic measures to aid AUV survey planning. Based on the bathymetric data that is available *a priori*, the measures capture the expected informativeness of an unseen environment, in terms of the expected *additional* information through in-situ visual observation. Experiments on both simulated data and real marine data demonstrate that the measures are able to predict the true utility of unobserved areas.

1.4 Outline

This thesis is structured as follows.

Chapter 2 establishes the background in feature learning, multimodal learning and benthic classification. The models described in this chapter are utilised and built upon in the following chapters.

Chapter 3 discusses the application of feature learning and classification techniques to marine data. After the marine datasets are introduced, various feature learning techniques are applied separately to the visual and bathymetric data modalities, and the ensuing classification results are presented.

Chapter 4 outlines a multimodal model based on stacked denoising autoencoders (DAEs) that learns the relationship between visual images and bathymetry. The

model is used to perform classification from various modality combinations, as well as habitat mapping tasks.

Chapter 5 extends the aforementioned model using a gated mixture of Restricted Boltzmann Machines, to better model the one-to-many mapping from bathymetric features to image features. Extensions to the original model are presented, to avoid having to specify the number of mixtures and to predict visual features given bathymetry as input. The model is used to cluster the input data (from either or both modalities), extract features for classification, and generate utility maps that can aid survey planning in unseen areas.

Chapter 6 proposes a number of information-theoretic measures to aid survey planning, based on the gated model described in Chapter 5. The measures are designed to predict the utility of acquiring visual image data in unobserved environments, given the bathymetric data over the region. The measures are used to rank a set of candidate dive locations, and to generate utility maps over a region of interest.

Chapter 7 concludes the thesis and suggests avenues for future work.

Chapter 2

Background

This chapter presents some background on unsupervised feature learning, deep learning, and classification of marine data. The models presented in this chapter are built upon in the following chapters of this thesis. Sections 2.1 and 2.2 present a review of the literature in semantic mapping and benthic habitat classification. Section 2.3 introduces the standard unsupervised feature learning techniques, and Section 2.4 builds on this to describe the commonly used deep learning models. Finally, Section 2.5 analyses the previous work in multimodal learning.

2.1 Semantic classification and mapping

A key task for many robotic vehicles is to categorise regions in its environment and build a semantic map of its surroundings. This capability allows an autonomous vehicle to perform high-level missions based on the objects and scenes that it encounters.

A number of methods perform semantic classification by combining laser and vision-based observations. Pronobis et al. [72] perform classification in an indoor office environment by utilising multiple visual and laser cues under a Support Vector Machine (SVM) framework. By combining these semantic labels with navigation information, the robot is able to generate a topological map indicating which room it is

in at each node in its pose graph. This work is then extended using a chain-graph to incorporate contextual information, such as adjacent class labels, into the process, allowing the robot to reason about unexplored areas [71]. Douillard et al. [21] utilise a model based on Conditional Random Fields (CRF) to capture spatial and temporal dependencies in the semantic mapping process. The semantic information extracted by these techniques can then be used for robot task planning [27].

However, while these techniques utilise both laser and visual information, they do not attempt to learn the relationship between the two modalities. There are numerous benefits to modelling the joint relationship between modalities, as previous approaches to multimodal learning have shown [40, 69, 85]. Firstly, we expect that including visual data at feature learning time leads to better remote sensing features, which enables more accurate, large-scale semantic classification. Secondly, such a model could then assign semantic meaning to its surroundings in an unsupervised fashion, by extracting key features and clustering the environment. Lastly, visual information could be predicted or inferred in unseen areas from the remotely sensed data, which enables multimodal queries about the environment in areas where one of the modalities is unavailable.

2.2 Benthic habitat classification

AUVs are often deployed to take high-resolution images of the seafloor along with numerous other sensor measurements, such as temperature, salinity and conductivity [22, 62, 82, 99, 103]. While in-situ observation can also be performed with towed camera sleds or diver rigs equipped with sensor suites [13, 91], AUVs offer a number of advantages. Specifically, they can autonomously follow the ocean floor at fixed altitudes, even for rugged terrain, and are far less constrained than human divers in terms of survey depth and duration [6]. In addition to in-situ measurements from any of these platforms, there is also a wealth of remotely sensed data available, most commonly in the form of bathymetry (ocean depth) and backscatter (reflectance) data from shipborne multi-beam sonar [84].

Semantic classification techniques can be applied to this data to generate *benthic habitat maps*, which classify large regions of the seafloor into broad habitat classes based on their physical and biological constituents [98]. These habitat maps are invaluable data products to marine scientists, assisting in monitoring the distribution and health of various benthic species [82, 99, 103]. Moreover, this semantic understanding of the benthos facilitates the long term autonomy of an AUV, allowing it to perform exploration missions in line with a high-level goal (e.g. "find and monitor kelp forests").

Benthic habitats are primarily determined by the substrate (such as rock or sediment) and the organisms present (such as algae or coral) [46], making them relatively easy to distinguish using in-situ image data. As a result, various techniques perform habitat classification using visual imagery, by performing supervised classification of coral reef survey images [4, 61], or clustering benthic imagery in an unsupervised fashion [29, 88]. Some approaches are also able to perform semantic mapping in real-time on board the vehicle [31, 42]. However, AUVs can only traverse a tiny fraction of a larger area of interest, limiting the scale to which visual habitat classification can be performed. Conversely, acoustic data is usually available a priori over an entire site, but has a low spatial resolution, on the order of metres between adjacent readings. Given this tradeoff, large-scale habitat mapping methods tend to be based on multibeam acoustic bathymetry or backscatter data, with the visual imagery acting as "ground truth" [14, 44]. In fact, many AUVs are equipped with a multibeam sonar [49], and the resulting high resolution bathymetry and backscatter maps can be used for habitat mapping, but this is again restricted by the limited coverage of the AUV.

The relationship between the topography of the seafloor and the presence of different benthic species is well documented in the literature [2, 46, 56], with terrain complexity being a strong indicator for the presence of some habitat classes and species [47]. Four bathymetric features that are key to determining the underlying habitat are (1) the depth; (2) the rugosity, or ruggedness of the surrounding terrain; (3) the slope; and (4) the aspect, or direction of greatest slope [14, 57, 100]. Friedman et al. [26] describe techniques based on Principal Component Analysis (PCA) to extract these features

in an unsupervised fashion from dense 3D reconstructions of the seafloor derived fom stereo visual imagery. Bender et al. [5] extract these features at multiple scales from shipborne bathymetric data, and also incorporate visual information into the process by clustering AUV-based benthic imagery: the probabilistic cluster assignments are used as training labels for bathymetric classification. Another method extrapolates vision-based results to larger regions, using visual classification from a completed dive to determine the most informative future dive from a set of candidates [77].

Acoustic backscatter, or the intensity of the sonar return, captures the reflective properties of the substrate, and can therefore also be a strong indicator of the underlying habitat class [15, 25, 56, 78]. However, it is also modulated by parameters unrelated to the benthic habitat, such as the beam incidence angle, range and footprint size, which results in noisy artefacts such as nadir and outer beam effects [28]. Consequently, extensive processing is usually performed on the backscatter data mosaics to correct for these effects [14]. Nonetheless, numerous contemporary studies make use of both bathymetry and backscatter mosaics for benthic habitat characterisation [39, 75]. While bathymetry is also susceptible to noise and hence requires postprocessing, backscatter artefacts appear more strongly in the Tasmanian dataset and require additional modelling effort. Since the focus of this thesis is on multimodal learning, the use of backscatter data is left as a future research direction (Chapter 7), and the focus is on utilising the bathymetry, or topographical structure, of the seafloor.

Building on these techniques, the approach proposed in this thesis looks to incorporate both bathymetric and visual features into the classification process, whilst maintaining the ability to classify either modality on its own.

2.3 Unsupervised Feature Learning models

This section describes a number of unsupervised feature learning models that are commonly used in the literature. The focus is on single layer feature learners, with the aim of extending these to deep models in the following section.

2.3.1 Overview

Feature learning refers to a family of learning techniques that attempt to determine a set of basis vectors or features to describe a dataset, often with a sparse representation. Different algorithms can perform feature learning in practice, including autoencoders, k-means clustering, Gaussian mixture models and restricted Boltzmann machines (RBMs) [17]. These methods all tend to learn similar dictionaries of localised filters [17], such as Gabor-like edge filters for natural images, or handwriting "strokes" for the MNIST digits dataset. While RBMs are generative models that can sample from the data-generating distribution [35], autoencoders are trained to optimise their reconstruction of the input data.

2.3.2 Autoencoders

An Autoencoder (AE) is a single layer neural network in which the hidden layer learns to reconstruct the input. The input $\mathbf{x} \in [0,1]^{n_x}$ is *encoded* to a hidden layer representation $\mathbf{h} \in [0,1]^{n_h}$, which is then *decoded* to an output $\mathbf{x}' \in [0,1]^{n_x}$. This $\mathbf{x}'$ represents the *reconstruction* of the input $\mathbf{x}$, and by training the network to minimise the difference between the two, the model learns a mapping to a feature representation $\mathbf{h}$ that is able to reconstruct the input data (Figure 2.1).

The encoding and decoding equations are given by:

$$\begin{aligned} h_j &= \text{sigm}\left(b_j + \sum_{i=1}^{n_x} w_{ij} x_i\right) \\ x'_i &= \text{sigm}\left(a_i + \sum_{j=1}^{n_h} w'_{ij} h_j\right) \end{aligned} \tag{2.1}$$

Here, n_x and n_h are the dimensionality of the input and hidden representations, $\text{sigm}(x) = \frac{1}{1+e^{-x}}$ is the element-wise logistic sigmoid function, $\mathbf{W} = [w_{ij}]$ and $\mathbf{W}' = [w'_{ij}]$ are weight matrices, and $\mathbf{a} = [a_i]$ and $\mathbf{b} = [b_j]$ are bias vectors.

In the case of real-valued data $\mathbf{x} \in \mathbb{R}^d$, a linear decoder $x'_i = a_i + \sum_j w'_{ij} h_j$ is usually

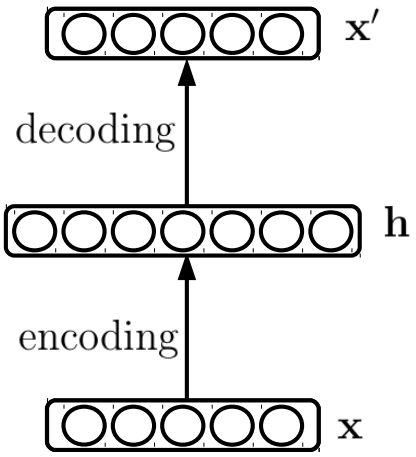


Figure 2.1 – Graphical representation of an autoencoder. The model is trained to minimise a loss function between the input $\mathbf{x}$ and the reconstruction $\mathbf{x}'$.

used for the reconstruction. The model parameters are often further constrained by using *tied weights*, $\mathbf{W}' = \mathbf{W}^\top$ [17]. This acts as a regulariser and affords additional flexibility in the model, such as the option to fine tune the model as an RBM.

Given a training set of N input data vectors, each training vector $\mathbf{x}^{(n)}$ can be mapped to a hidden representation $\mathbf{h}^{(n)}$, followed by reconstruction $\mathbf{x}'^{(n)}$. The model parameters $\Theta = \{\mathbf{W}, \mathbf{a}, \mathbf{b}\}$ are then tuned to minimise a loss function, often the mean squared reconstruction error over the training set:

$$\begin{aligned} J(\theta) &= \frac{1}{N}\sum_{n=1}^{N} \left\|\mathbf{x}^{(n)} - \mathbf{x}'^{(n)}\right\|_2^2 \\ \theta^* &= \underset{\theta}{\operatorname{argmin}}\ J(\theta) \end{aligned} \tag{2.2}$$

Typically, the parameters are learned using Stochastic Gradient Descent (SGD) or another gradient-based optimisation procedure. As a result, the autoencoder learns a hidden layer representation to minimise the mean squared error between the input

and the model-based reconstruction.

A number of modified versions have also been introduced in the literature, including the contractive autoencoder [76], which learns features that are invariant to perturbations in the input space; the variational autoencoder [45], which provides efficient variational methods for training and generative inference; and an online incremental autoencoder that is able to add or merge hidden units on-the-fly based on a continuous data stream [105].

2.3.2.1 Regularisation and Sparsity

To prevent the weights from increasing unboundedly, and to improve generalisation on unseen data, a regularisation term is often added to the loss function. Typically, this is the L_2 weight decay term, the square of the L_2 norm of weight matrix $\mathbf{W}$. This has the effect of shrinking the weights that are less useful in the reconstruction process. Another common option is L_1 weight decay, which has the effect of setting redundant weights to zero.

Further, hidden units that are selectively activated have been shown to be more useful in discriminative tasks [17]. As a result, it is also common to incorporate a sparsity cost, based on the cross entropy between the sparsity (average activation) of each unit, $\hat{\rho}_j = \frac{1}{N}\sum_{n=1}^{N} h_j^{(n)}$, and a user-defined sparsity ρ.

The entire objective function, including weight decay and sparsity cost, is given by:

$$J(\theta) = \frac{1}{N}\sum_{n=1}^{N} \|\mathbf{x}^{(n)} - \mathbf{x}'^{(n)}\|_2{}^2 + \lambda \|\mathbf{W}\|_F{}^2 + \beta \sum_{j=1}^{n_h} \left[\rho \log \frac{\rho}{\hat{\rho}_j} + (1-\rho)\log \frac{(1-\rho)}{(1-\hat{\rho}_j)}\right] \tag{2.3}$$

Here, λ and β are hyperparameters to tune the effects of weight decay and sparsity cost, respectively.

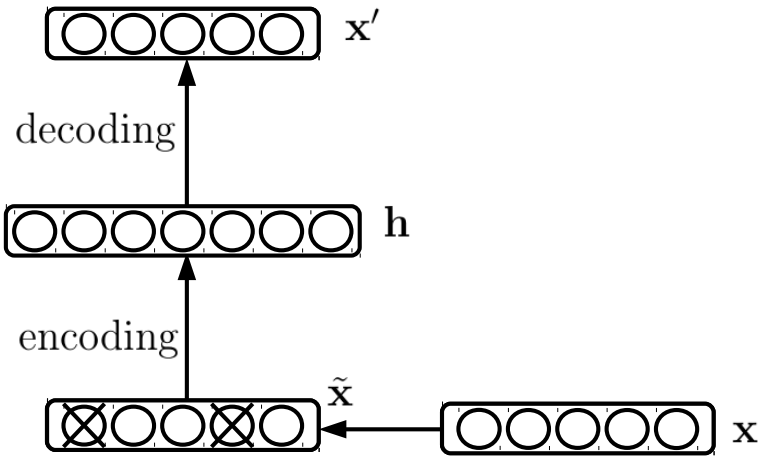


Figure 2.2 – Graphical representation of a denoising autoencoder. In this case, masking noise is applied to the input data. The model is trained to minimise a loss function between the *clean* input $\mathbf{x}$ and the reconstruction $\mathbf{x}'$

2.3.3 Denoising Autoencoders

Another way to regularise an autoencoder model is to apply a stochastic corruption $q(\tilde{\mathbf{x}} \mid \mathbf{x})$ to each data vector $\mathbf{x}^{(n)}$ during training. The corrupted vector $\tilde{\mathbf{x}}^{(n)}$ is then used as the training input, but the loss function compares the model reconstruction with the *clean* input (Figure 2.2). As a result, this Denoising Autoencoder (DAE) [93] learns to reconstruct input data with robustness to corruption / noise. In other words, it learns a set of features that can undo noisy perturbations to reconstruct the clean input.

Typical options for the stochastic corruption include masking noise or additive isotropic Gaussian noise. In the case of masking noise, a fraction η of the input dimensions are set to zero, and the model learns features that are robust to missing input dimensions.

The corruption process is stochastic, so the noise applied varies for each training vector and for each iteration of learning. However, after training the model, the hidden representation is obtained using *clean* inputs, so that future tasks with the encoded features are not probabilistic.

2.3.4 Restricted Boltzmann Machines

A Restricted Boltzmann Machine (RBM) is a stochastic generative neural network comprised of a set of binary visible variables $\mathbf{x} \in \{0,1\}^{n_x}$ and binary hidden variables $\mathbf{h} \in \{0,1\}^{n_h}$. The joint distribution $p(\mathbf{x}, \mathbf{h})$ is specified by an energy function:

$$\begin{aligned} E(\mathbf{x}, \mathbf{h}) &= -\sum_i a_i x_i - \sum_j b_j h_j - \sum_{ij} w_{ij} x_i h_j \\ p(\mathbf{x}, \mathbf{h}) &= \frac{e^{-E(\mathbf{x},\mathbf{h})}}{Z} \end{aligned} \tag{2.4}$$

Here, $\mathbf{W} = [w_{ij}]$ is the weights matrix, $\mathbf{a} = [a_i]$ and $\mathbf{b} = [b_j]$ are the visible and hidden bias vectors respectively, and $Z = \sum_{\mathbf{x},\mathbf{h}} e^{-E(\mathbf{x},\mathbf{h})}$ is the partition function.

An RBM can be described using the concept of a *probabilistic graphical model*, which utilises a graph-based representation to express the dependences between random variables. In an RBM, the visible and hidden units form a bipartite graph. That is, the visible units are all independent when conditioned on the hidden units, and vice versa. This conditional independence property yields the following familiar conditional expressions:

$$\begin{aligned} p(h_j = 1|\mathbf{x}) &= \text{sigm}\left(b_j + \sum_i w_{ij} x_i\right) \\ p(x_i = 1|\mathbf{h}) &= \text{sigm}\left(a_i + \sum_j w_{ij} h_j\right) \end{aligned} \tag{2.5}$$

where $\text{sigm}(x) = (1 + e^{-x})^{-1}$ is the element-wise logistic sigmoid function.

The graphical representation of an RBM is shown in Figure 2.3. The parameter, input, and hidden spaces are all identical to the autoencoder.

The probability of an input vector $\mathbf{x}$ can be obtained by marginalising the joint density $p(\mathbf{x}, \mathbf{h})$ over the hidden units:

$$F(\mathbf{x}) = -\sum_i a_i x_i - \sum_j \log\left(1 + e^{b_j + \sum_i w_{ij} x_i}\right)$$

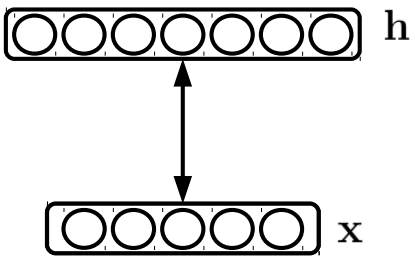


Figure 2.3 – Graphical representation of a Restricted Boltzmann Machine.

$$p\left(\mathbf{x}\right) = \frac{\sum_{\mathbf{h}} e^{-E(\mathbf{x},\mathbf{h})}}{Z} = \frac{e^{-F(\mathbf{x})}}{Z} \tag{2.6}$$

where the expression $F\left(\mathbf{x}\right)$ is known as the *free energy* of a visible vector. Unfortunately, the partition function Z is intractable, which means that the RBM can only compute *unnormalised* probabilities. However, several techniques in the literature can approximate the partition function if necessary [79].

A number of previous works have introduced variants of the standard RBM model, including the Gaussian RBM [96, 97], which is similar to using a linear decoder in an autoencoder; the discriminative RBM [51], which extends the RBM to a supervised model; and the spike-and-slab RBM [19], which utilises both a binary *spike* variable and a real-valued *slab* variable for each of the hidden units.

2.3.4.1 Training

Given a set of training vectors $\{\mathbf{x}^{(1)}, \cdots, \mathbf{x}^{(N)}\}$, RBM models are usually trained to maximise the mean log probability of the data, $L = \frac{1}{N}\sum_{n=1}^{N} \log p(\mathbf{x}^{(n)})$ with respect to the parameters Θ, using Stochastic Gradient Descent. The gradient term is given by:

$$\frac{\partial L}{\partial \Theta} = N\mathbb{E}\left[\frac{\partial E(\mathbf{x},\mathbf{h})}{\partial \Theta}\right] - \sum_{n=1}^{N} \mathbb{E}\left[\frac{\partial E(\mathbf{x}^{(n)},\mathbf{h})}{\partial \Theta}\middle|\mathbf{x}^{(n)}\right] \tag{2.7}$$

The second term is a *data-driven* expectation, which can be estimated by using Gibbs

sampling to draw unbiased samples from the conditional distribution $p\left(\mathbf{h} \mid \mathbf{x}^{(n)}\right)$. However, the first term is a *model-driven* expectation and is intractable in practice, as it requires a sum over all $\mathbf{x}$ and $\mathbf{h}$. To sample from this distribution would require initialising the input dimensions randomly and performing alternating Gibbs sampling for a very long period of time.

As a result, the Maximum Likelihood gradients are approximated using the Contrastive Divergence (CD) algorithm [36], commonly used for a variety of energy-based models. The key approximation is to initialise the Gibbs chain at the value of a training vector rather than at random values when computing the model-driven expectation. If we consider that the visible and hidden nodes form a Markov chain, this ensures that the chain is 'close' to the stationary distribution and fewer iterations of Gibbs sampling are required (typically only one).

The procedure is shown in Algorithm 2.1. For a batch of data, the first step of the algorithm is to sample the hidden variables $\mathbf{h}_+$ from the input $\mathbf{x}_+$. This is known as the *positive phase*, and the input and hidden data represent the data-driven statistics. Next, the model reconstruction $\mathbf{x}_-$ is sampled from the hiddens, to complete a single iteration of Gibbs sampling. Multiple iterations of Gibbs sampling can be executed (CD-n), but a single iteration is often sufficient (CD-1). Finally, $\mathbf{x}_-$ is used to sample $\mathbf{h}_-$, representing the *negative phase* of training, or the model-driven statistics. The CD algorithm then approximates the gradients with a difference between the data-driven statistics and model-driven statistics. The computed gradient is likely to be small if the model's representation is similar to the data-driven representation, or large if otherwise.

2.3.5 The connection between AEs and RBMs

Clearly, there are a number of similarities between autoencoders and RBMs. For both models, the encoding function from inputs to hidden units requires a linear projection and nonlinear activation function. The decoding functions are also identical if the autoencoder is trained with tied weights (i.e. the decoding weights are the transpose

Algorithm 2.1: Contrastive Divergence (CD-1) training for RBMs

$\frac{\partial L}{\partial \mathbf{W}} \Leftarrow \mathbf{0}$, $\frac{\partial L}{\partial \mathbf{a}} \Leftarrow \mathbf{0}$, $\frac{\partial L}{\partial \mathbf{b}} \Leftarrow \mathbf{0}$
for $i = 0$ to N **do**
  $\mathbf{x}_+ \Leftarrow$ training sample i
  Sample $\mathbf{h}_+ \sim p(\mathbf{h} \mid \mathbf{x}_+)$
  Sample $\mathbf{x}_- \sim p(\mathbf{x} \mid \mathbf{h}_+)$
  Sample $\mathbf{h}_- \sim p(\mathbf{h} \mid \mathbf{x}_-)$
  $\frac{\partial L}{\partial \mathbf{W}} \Leftarrow \frac{\partial L}{\partial \mathbf{W}} + \frac{1}{N}\left(\mathbf{x}_-\mathbf{h}_-^T - \mathbf{x}_+\mathbf{h}_+^T\right)$
  $\frac{\partial L}{\partial \mathbf{a}} \Leftarrow \frac{\partial L}{\partial \mathbf{a}} + \frac{1}{N}(\mathbf{x}_- - \mathbf{x}_+)$
  $\frac{\partial L}{\partial \mathbf{b}} \Leftarrow \frac{\partial L}{\partial \mathbf{b}} + \frac{1}{N}(\mathbf{h}_- - \mathbf{h}_+)$
end for

of the encoding weights).

Thus, an RBM can effectively be considered a probabilistic version of an autoencoder. The main trade-off between the two is the simplicity of the reconstruction error training criterion for the autoencoder (recall that Contrastive Divergence training for an RBM is approximate) versus the ability of the RBM to perform generative sampling tasks. As a result, the models are often used interchangeably in the literature.

However, a number of recent papers demonstrate the generative capabilities of autoencoder models, and uncover a stronger connection between Autoencoders and RBMs. Vincent et al. [94] illustrate that Autoencoders are able to generate plausible samples from the underlying data distribution when they are regularized by a denoising criterion, but not when regularized with a sparsity penalty. A more recent work [92] demonstrates that a Denoising Autoencoder with real-valued visible units and Gaussian input noise is equivalent to a Gaussian-Binary RBM trained under a different training criterion known as *Score Matching* [38]. More generally, autoencoders trained with Gaussian corruption under a mean-squared reconstruction error loss function capture the gradient of the log probability, or *score*, of the data [1]. Finally, Bengio et al. [8] generalise this to DAEs trained under an arbitrary reconstruction loss and corruption procedure, and propose methods to sample from such models.

As a result, Denoising Autoencoders can be considered as fully probabilistic models in

their own right [43]. While they do not model the marginal distribution over hidden variables [94], they can successfully sample from the underlying distribution as long as the visible units are initialised to an input data vector.

2.3.6 Other single layer learners

While AEs and RBMs form the basic building blocks of unsupervised deep learning models, there are various other algorithms that can perform single layer feature learning and encoding.

In general, the goal is to learn a set of basis vectors $\mathbf{D} = [\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_K]$, often referred to as a *codebook* or *dictionary*, and a representation $\mathbf{c}^{(n)}$ for each input vector $\mathbf{x}^{(n)}$, which represents a linear combination of the bases:

$$\mathbf{x}^{(n)} = \mathbf{D}\mathbf{c}^{(n)} \tag{2.8}$$

For a complete or undercomplete set of basis vectors (i.e. when the number of bases is less than or equal to the number of dimensions in the input data), a dictionary can be efficiently learned using Principal Components Analysis (PCA). PCA involves finding a linear transformation for the input data such that the dimensions of the transformed data are uncorrelated. The transformed feature dimensions are known as principal components, and form an orthogonal basis which captures the directions of highest variance in the input data [66]. In fact, the principal components can be computed directly as the eigenvectors of the covariance matrix of the input data, and ordered by eigenvalue (representing the variance of each component). Then, the undercomplete basis obtained by projecting onto a subset of these components has the property of preserving the maximum amount of variance in the data.

Often, however, an overcomplete basis is desired, where the number of bases is greater then the number of input dimensions. This can be advantageous because the bases are able to more accurately describe the structure present in the data. However, with an overcomplete basis, the linear coefficients $\mathbf{c}_i$ cannot be uniquely determined

from the input data, and the problem is degenerate. This is usually resolved by adding a requirement that the resulting representation be sparse, leading to a set of techniques known as *sparse coding* or *sparse dictionary learning* [18, 89]. Formally, these techniques look to solve the following constrained optimization problem:

$$\min_{\mathbf{D},\mathbf{C}} \|\mathbf{X} - \mathbf{DC}\|^2 \text{ s.t. } \forall n \ \|\mathbf{c}^{(n)}\|_0 \leq T \tag{2.9}$$

where $\mathbf{X} = \left[\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(N)}\right]$ and $\mathbf{C} = \left[\mathbf{c}^{(1)}, \mathbf{c}^{(2)}, \ldots, \mathbf{c}^{(N)}\right]$ are the data and associated representations. That is, the objective function seeks to minimise the $\mathcal{L}_2$ norm between the data and the dictionary-based reconstruction, subject to the constraint that the number of non-zero elements in each representation is bounded by some threshold T. If $T = 1$, this is known as *vector quantisation* [18].

One drawback of the above approach is that the $\mathcal{L}_0$ norm is very difficult to optimise, as it is non-convex. As a result, it is often replaced by the $\mathcal{L}_1$ norm, which is a good convex approximation [66], and is incorporated as a penalty term with a Lagrange multiplier rather than a hard constraint:

$$\min_{\mathbf{D},\mathbf{C}} \|\mathbf{X} - \mathbf{DC}\|^2 \ + \sum_{n=1}^{N} \lambda \|\mathbf{c}^{(n)}\|_1 \tag{2.10}$$

In fact, an $\mathcal{L}_1$ constraint forces elements to be exactly zero, resulting in sparse representations. This can be understood by the follow considerations. The solution for a constrained optimisation problem occurs at the point where the lowest level set of the loss function intersects the constraint surface [66]. The $\mathcal{L}_1$ constraint surface is a polytope centred at the origin, with its vertices along each axis. If we start with a tight constraint surface and relax it (making the surface larger), the vertices are much more likely to intersect with the loss than other points, meaning that the solution to the constrained optimisation problem is more likely to occur at these vertices, where several dimensions are equal to zero. For a more detailed, graphical explanation, the reader is directed to [9, 66].

2.4 Deep Learning models

Deep learning models have multiple layers of features within a single model. They are based on multi-layer neural networks, and each layer usually learns a set of features at a different scale or complexity.

This section outlines a number of deep models that are commonly used in the literature, including standard feedforward neural networks, deep belief networks (DBNs), and convolutional neural networks (CNNs).

2.4.1 Feedforward Neural Networks

A feedforward neural network involves multiple layers of hidden units / neurons, with the activations of each layer's neurons determined from the neurons in the layer below (Figure 2.4). The activation of a neuron in layer k is a linear mapping of the neuronal activations of layer $k-1$, followed by a nonlinear squashing function, often a sigmoid. As such, the mapping from one layer to the next is equivalent to the encoding phase of an autoencoder.

Given an input $\mathbf{x}$ to the network (training or test data), the network can compute the value of the output units, which usually represent a structured output such as probabilities over a set of classes. The network is then trained to minimise the error between the predicted output $\mathbf{y}$ and ground truth labels $\mathbf{y}_t$, using a gradient descent approach. This involves computing the gradient of the error term with respect to the parameters of each layer, a procedure known as backpropagation (i.e. propagation of errors back through the network).

As an example, suppose we have a feedforward network with n hidden layers and an output layer, each composed of sigmoid units. The loss function is a mean squared error cost $J = \frac{1}{2}(\mathbf{y} - \mathbf{y}_t)^2$, and the activations of each layer are given by:

$$\begin{aligned} \mathbf{z}^{(k+1)} &= \mathbf{W}^{(k)}\mathbf{h}^{(k)} + \mathbf{b}^{(k)} \\ \mathbf{h}^{(k+1)} &= \sigma(\mathbf{z}^{(k+1)}) \end{aligned} \tag{2.11}$$

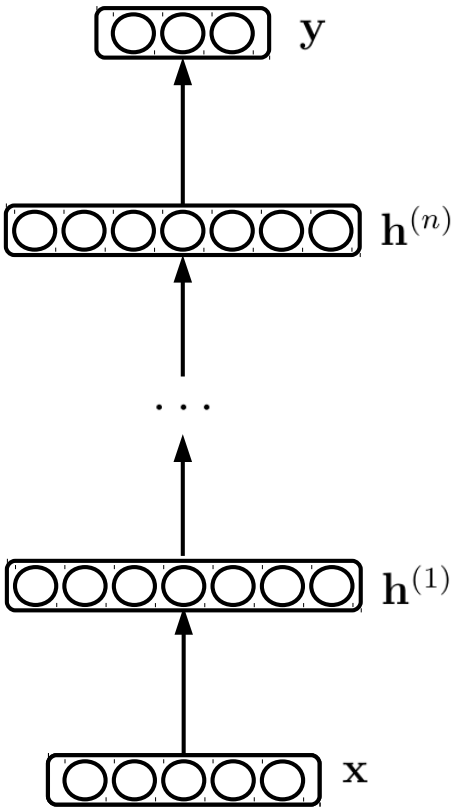


Figure 2.4 – Graphical representation of a feedforward neural network.

where $\mathbf{W}^{(k)}$ and $\mathbf{b}^{(k)}$ are the weights and biases of the k^{th} layer. Using the chain rule, the gradient of the objective function with respect to the activations of each layer are given by:

$$\begin{aligned} \frac{\partial J}{\partial \mathbf{z}^{(n)}} &= \boldsymbol{\delta}^{(n)} = (\mathbf{y} - \mathbf{y}_t) \cdot \sigma'(\mathbf{z}^{(n)}) \\ \frac{\partial J}{\partial \mathbf{z}^{(k)}} &= \boldsymbol{\delta}^{(k)} = \left((\mathbf{W}^{(k)})^T \boldsymbol{\delta}^{(k+1)}\right) \cdot \sigma'(\mathbf{z}^{(k)}) \end{aligned} \tag{2.12}$$

It can be seen that the error gradient is propagated back through the layers of the network, with the gradient with respect to one layer being computed from the gradient with respect to the layer above. The final gradients with respect to each of the

parameters can be calculated as:

$$\begin{aligned} \frac{\partial J}{\partial \mathbf{W}^{(k)}} &= \boldsymbol{\delta}^{(k+1)}(\mathbf{h}^{(k)})^T \\ \frac{\partial J}{\partial \mathbf{b}^{(k)}} &= \boldsymbol{\delta}^{(k+1)} \end{aligned} \tag{2.13}$$

While feedforward neural networks were traditionally trained using this backpropagation procedure, it is susceptible to the "vanishing gradients" problem, whereby the gradient of the error term becomes increasingly smaller with respect to the parameters of the lower layers. As a result, neural networks were typically limited in the number of layers used, until unsupervised layer-wise training approaches were introduced [34].

2.4.2 Deep Belief Networks

Deep Belief Networks (DBNs) are composed of multiple layers of unsupervised feature learners stacked into a deep architecture. They are trained layer-by-layer in an unsupervised fashion, by training an RBM on the input data, obtaining the hidden layer representation, and then using this as the input to the next layer RBM. This layer-wise unsupervised training procedure can be used to initialise the parameters of a feedforward neural network before performing supervised training ("fine tuning"). It is believed that unsupervised pre-training acts as a regulariser in supervised training: the model parameters are initialised closer to a good local minimum for supervised tasks, within a basin of attraction that corresponds to parameters also useful for unsupervised tasks [24]. Effectively, greedy layer-wise training avoids the problem of vanishing gradients and has led to deep networks achieving state of the art performance in a range of learning and classification tasks [32, 52, 80].

A single layer autoencoder may also be used to train each layer, in which case the model is often termed a *deep autoencoder*. The weights of each layer's autoencoder are usually tied during the layer-wise training phase, but are then untied in order to "unroll" the model into a single multi-layer autoencoder. Recent techniques have also enabled DBNs to be trained jointly, without layer-wise training [65].

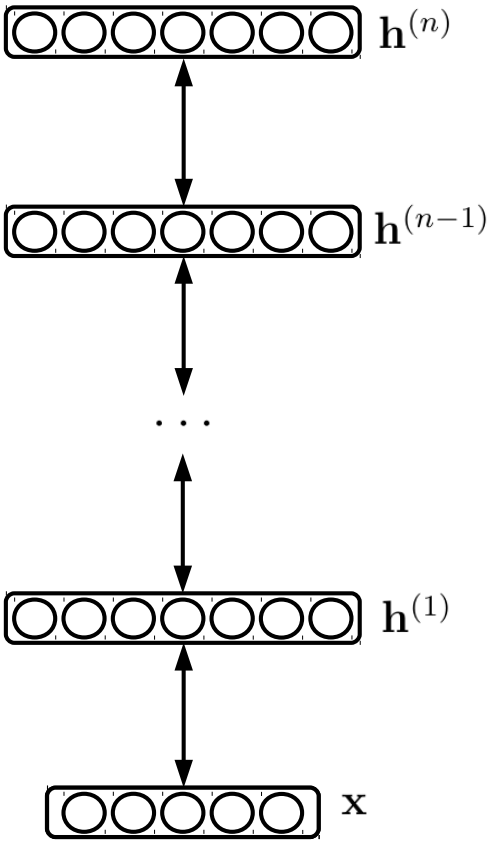


Figure 2.5 – Graphical representation of a Deep Belief Network.

Deep networks are able to capture high-level features in an input dataset. Each layer of a deep model learns a progressively higher order correlation in the input dataset, which often corresponds to a higher level of feature abstraction. When trained on natural images, such models can learn entire hierarchies of features: edges, combinations of edges, object parts, and entire object templates [52, 73]. Lee et al. [53] demonstrate that the hierarchical structure learned by these models mimics the neural activities of area V2 in the visual cortex of the human brain. It has also been shown that each neuron in the top layer can capture a significant factor of variation in the data that corresponds to a single qualitative characteristic. Cheung et al. [16] train a deep generative model on images of human faces, and demonstrate that many of the individual features capture characteristics such as facial shape or key emotions such as joy or anger. In fact, by manually changing the activations of the top layer,

they are able to artificially modify the faces to express emotion to differing degrees.

2.4.3 Convolutional Neural Networks

Until this point, we have only considered *fully connected* models: that is, models in which neurons are connected to all of the input dimensions. If we consider a typical image classification or object recognition task, the input data is usually in the form of large visual images. Even for a modest size of 128×128 pixels for an input image, this results in 16384 input dimensions, which can be prohibitive for even the simplest of networks.

Convolutional Neural Networks (CNNs) offer a solution to this problem. Instead of each hidden unit connecting to all of the pixels in the input image, it is only connected to pixel positions within a local patch, known as its *receptive field*. The weights for a particular hidden unit are then shared for all positions in the input image. As such, this acts as a *convolutional layer*, with a local filter being convolved over an entire image to produce a feature map.

CNNs are usually composed of several such convolutional layers, separated by pooling layers. The convolutional layers apply the local filter to all positions in the image, while the pooling layers reduce the size of the encoded data by downsampling the resulting feature map. The most common type of pooling used is max pooling, which outputs the maximum value over each pooling region, but mean pooling is also used in some of the literature. These pooling layers are particularly important because they reduce the computation for higher layers, by removing non-maximal hidden unit activations. They also act as a form of translational invariance: by pooling over a 2×2 region, for example, a maximal activation can translate by one pixel and still produce an identical output.

Following a series of alternating convolution and pooling layers, a number of fully connected layers may also be incorporated, to learn the high order correlations in the features. Fully connected layers are now feasible in the higher layers, as the input dimensionality has been significantly reduced through pooling. Finally, a multi-class

logistic regression, or *softmax*, classifier layer maps the top layer features to the image labels.

2.4.3.1 Dropout

One drawback of the fully connected layers in convolutional networks is that they are prone to overfitting [86]. Whereas the lower layer features are constrained by having to look at a small receptive field, and fully connected layers for other models are 'regularised' by performing layer-wise training, the fully connected layers in a convolutional network afford enough expressive power for the model to overfit the data.

As a result, the *dropout* method is usually applied to these layers [86]. Dropout is a simple model averaging technique that efficiently combines an exponential number of hidden layer architectures, each sharing the same set of weights. During training, for each input sample, each hidden unit in the fully connected layer is removed from computation ("dropped out") with a certain probability (usually $p = 0.5$). This means that it is unused in both the forward-pass and backpropagation stages. Following training, inference can then be performed with an approximate model averaging technique: all of the units are used in the encoding process, with the weights of each neuron scaled by $1 - p$ (the expected value of the number of units that remain during dropout training).

Dropout provides a number of benefits. Firstly, it prevents features from "co-adapting" to capture a particular feature in the input data: with hidden units dropping out randomly, any unit cannot rely on another feature being active. This process thereby ensures that each hidden unit is independent and robust, learning a feature that is useful in conjunction with the random subset of other features that is selected during dropout. Secondly, it can be considered a form of adaptive regularisation [95], leading to better generalisation on unseen data. Lastly, it has been shown that model combination nearly always improves the performance of machine learning models. By using the proposed encoding technique, the dropout model approximately averages

the predictions of exponentially many different models.

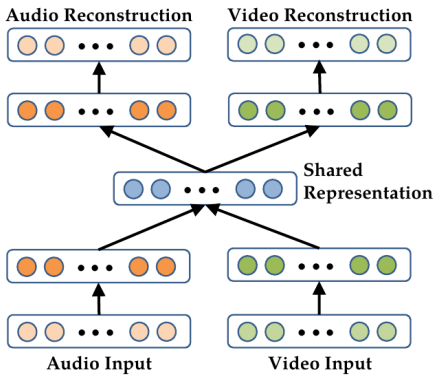
2.4.4 Applications

As a result of their enormous expressive power, deep learning methods have attained state-of-the-art performance in a range of tasks.

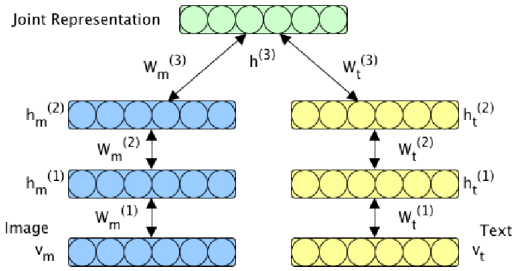
Graves et al. [32] apply deep recurrent neural networks to the task of speech recognition, utilising the Long short term memory (LSTM) unit [37], which is well suited to modelling time series data. Lee et al. [54] utilise a convolutional DBN to perform unsupervised feature learning on audio spectrogram data, and demonstrate that the learned features closely match the spectrograms of phonemes. As a result, they are able to perform a wide range of classification tasks, including phoneme classification, speaker classification, speaker gender classification, and music genre and artist classification.

One work looks at the problem of collaborative filtering in the context of the Netflix prize [80]: making movie recommendations for users based on (incomplete) information on the preferences and tastes of other users. In particular, they utilise a conditional RBM, and propose techniques to perform learning and inference when data dimensions are missing. This approach allows them to effectively utilise the sparse Netflix user recommendation data, leading to better perfomance than Netflix's own system.

A large body of work has investigated the use of deep learning techniques in computer vision applications. A number of previous papers [52, 74] utilise convolutional DBNs on visual image data, and both demonstrate the ability to learn hierarchies of features, from edges / gradients in the first layer, to combinations of edges in the second layer, to object parts and whole objects in the higher layers. Ranzato et al. [74] use this type of model as an invariant feature extractor for the image, and perform object classification on the MNIST digits dataset and the Caltech-101 objects dataset. Lee et al. [52], on the other hand, propose a fully probabilistic model, that can not only handle such classification tasks, but also complete an image that has been



(a) Deep autoencoder for audio-video learning [69]



(b) Deep Boltzmann Machine for image-keywords learning [85]

Figure 2.6 – Examples of deep multimodal architectures used in previous work.

corrupted. By demonstrating this ability, they show that convolutional DBNs can learn the underlying structure of the visual image data. Another work [68] uses a convolutional DBN with a third order Boltzmann machine, to perform 3D object recognition. Finally, Krizhevsky et al. [48] proposes a very large, very deep CNN which comprehensively achieves state-of-the-art classification performance on both the ImageNet dataset and in the ILSVRC-12 competition. They utilise a multi-GPU architecture, and propose a number of architectural modifications to make learning more efficient.

Deep learning techniques have also been applied to other interesting applications, such as the detection of grasps for robotic manipulation [55]; reward function estimation for reinforcement learning [64, 101]; and modelling human motion [90].

2.5 Multimodal learning

Deep learning techniques have previously been used for multimodal learning, since they are able to capture high-level correlations between two data modalities. Typically, this involves training a deep network for each modality separately, and then

training a multimodal layer on the concatenation of the high-level single modality features.

Ngiam et al. [69] use a deep learning approach to perform classification of phonemes from audio and video features. They train the model in a layer-wise fashion, but then fine tune it as a deep autoencoder (Figure 2.6(a)). They show that better features can be learned for one modality if both are used at feature learning time (shared representation learning), and demonstrate the ability to train a classifier on one modality and test on another (cross modality learning). Performing experiments with different architectures, they demonstrate that it is optimal to train a deep network on each modality separately followed by a single multimodal layer on top: this is because the types of multimodal correlations that exist are much more likely to be related to high-level concepts (such as words or phonemes) rather than lower level inputs.

Other papers learn the correlations between a dataset of images and associated keyword tags. One technique accomplishes this with a Deep Boltzmann Machine [85] (Figure 2.6(b)): by maintaining the generative properties of the RBM, the model can perform a range of inference tasks, such as classification, content-based image retrieval, and the ability to sample one modality from the other. Another approach uses a Bayesian co-clustering algorithm to learn a relationship between a visual dictionary and textual words, in order to perform classification and keyword-based image retrieval tasks [40].

Sohn et al. [83] propose training a multimodal model to minimise the *variation of information*, a measure of distance between the two modalities. They argue that this training objective better equips the model to predict missing modalities, which leads to state-of-the-art performance in image keyword annotation. Finally, Mao et al. [60] extend the problem to the annotation of images with full sentence descriptions. As such, they utilise a recurrent neural network to model the sentence structure, and a deep CNN to model the image content, with a multimodal layer to capture the relationship between the high-level features of each modality.

While these techniques span a variety of different architectures and data modalities, none are directly applicable to the task of multimodal learning from marine data, for

a number of reasons. Firstly, marine images are visually very different to the image datasets that have previously been used, which typically focus on objects or urban / outdoor scenes. Secondly, feature learning techniques have yet to be applied to acoustic bathymetry data, necessitating a new approach. And finally, the types of high-level correlations that exist between these two are likely to be very different to those between, for example, images and textual keywords.

The surveyed methods do, however, have one characteristic in common, in that they all utilise a shared multimodal layer to model the joint relationship between two modalities. Regardless of the models used to extract features from each modality, the shared layer is able to capture the correlations between these high-level features. This type of approach will be adopted throughout this thesis.

2.6 Summary

This chapter has summarised the literature in semantic classification and mapping from marine data, deep learning, and multimodal learning. Deep models have previously been applied to multimodal learning tasks, and are particularly well-suited to the problem because of their enormous expressive power and ability to capture high-level correlations in the underlying data. In particular, by learning high-level features on each modality separately and capturing cross-modality correlations using a shared representation layer, the model can work with whichever modalities are available at inference time. The following chapters build on previous work in order to solve the problem of multimodal learning from visual and bathymetric data.

Chapter 3

Learning features from marine data

This chapter investigates the application of various feature learning techniques to marine data, in terms of both visual images and acoustic multi-beam bathymetry (depth) data. The learned features are analysed, and compared to traditional hand-picked features that are typically used for classification tasks. The features are then applied to the task of classifying benthic habitats, using a variety of standard supervised classification algorithms. The effectiveness of each feature learning approach is gauged by its classification performance.

This chapter is arranged as follows. Section 3.1 describes the datasets used in this thesis, including the AUV-borne in-situ visual images, and remotely sensed bathymetry data. Section 3.2 describes the setup for the main classification problem of interest, and outlines the classifier and validation techniques used throughout this thesis. Section 3.3 details the bathymetric feature learning technique and compares it to traditional hand-selected features, presenting both classification and habitat mapping results. Section 3.4 outlines the algorithms used to extract features from visual imagery, and presents and compares classification results for these techniques.

3.1 Datasets

The main dataset used in this thesis is the Southeastern Tasmania dataset [5, 84, 98], which was acquired in 2008 based on a collaboration between The University of Sydney, The University of Tasmania, and Geoscience Australia. The following sections describe the dataset and how it was obtained.

3.1.1 Bathymetry

Bathymetry is the study of ocean topography, and refers to the depth of oceans and other large water bodies. Bathymetric data is usually acquired using shipborne sonar, in the form of either a Multi-beam Echosounder (MBES) or a sidescan sonar. In the case of a MBES, the underside of the ship is equipped with a transmitter and receiver. A series of sound pulses ('pings') are transmitted and received by the sonar head over a swath width of approximately four times water depth. Each pulse is reflected off the ocean floor, and is subequently detected by the receiver. Based on the time elapsed between transmitting and receiving each pulse, the range to each point on the ocean floor can be calculated. By pinging the seafloor at regular intervals, an area of seafloor corresponding to the swath width can be mapped while the ship is in forward motion.

Typically, the data is processed after acquisition, by removing outliers and combining co-located observations. The resulting bathymetric data product is in the form of a 2.5D Digital Elevation Map (DEM), which specifies the ocean depth at each point over a two-dimensional grid. In this thesis, the bathymetric data is in the form of large-scale gridded data from Geoscience Australia [84], as shown in Figure 3.1. The uniform grid has a separation of 1.6 m between grid points, and covers a depth range of $5 - 104$m. The grid was obtained by postprocessing bathymetric data collected by the *Challenger* research vessel in 2008 using a Simrad EM3002(D) 300kHz MBES system [84].

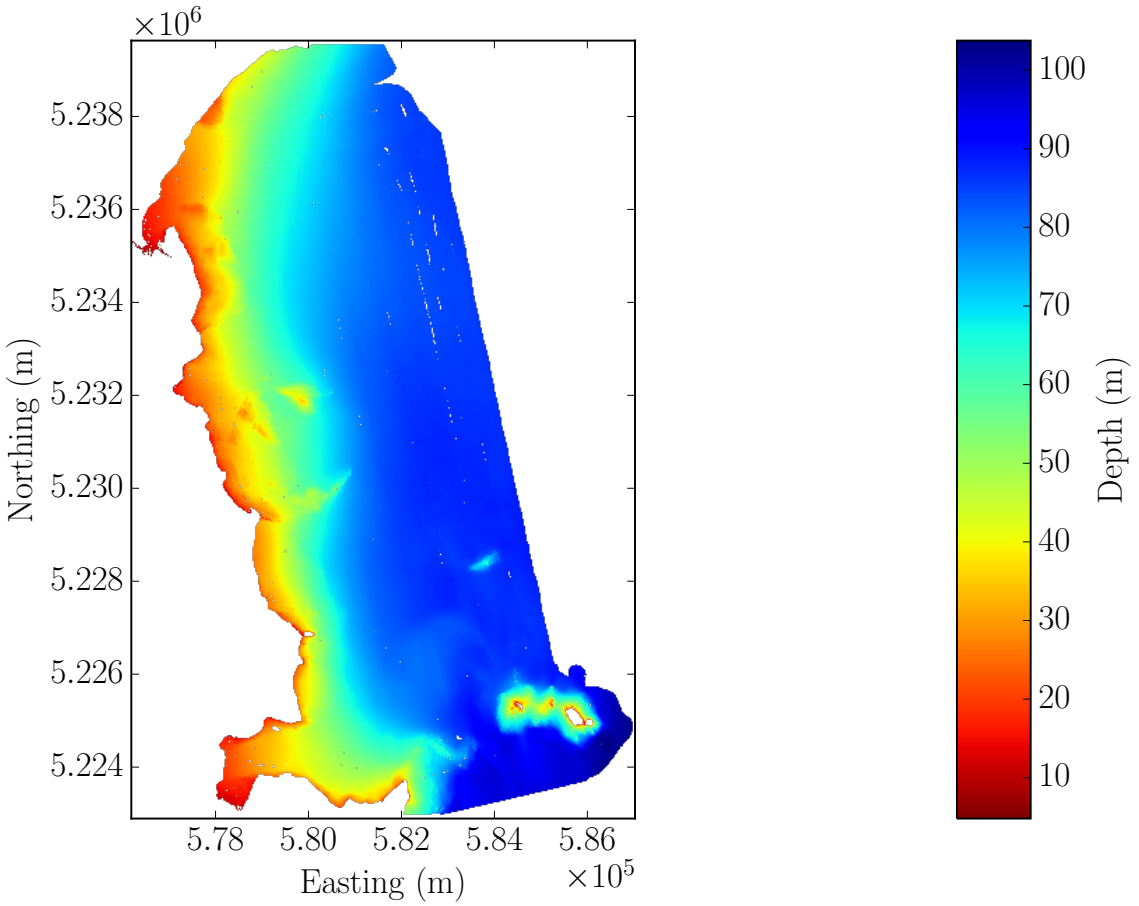


Figure 3.1 – The gridded bathymetry data over the entire Southeastern Tasmania region. The depth ranges from 5m (red) to 104m (blue). The black trajectories indicate the extent of the AUV surveys performed over the course of two weeks in 2008.

3.1.2 Visual Images

The visual image dataset is comprised of a large set of high-resolution photographs of the ocean floor, taken by a set of downward facing stereo cameras mounted on the AUV *Sirius* [98]. The 1360 × 1024 pixel images are spread over 11 different dives, covering a range of habitats from flat-bottomed sandy areas to kelp forests. As shown in Figure 3.1, the AUV surveys cover only a tiny fraction of the Southeastern Tasmanian shelf.

Most of the AUV images are taken at an altitude of approximately 2m above the seafloor. Images taken at a higher altitude tend to be very dark, while images at a

low altitude are extremely white. As a result, images outside the range of $1.5 - 3$m have been removed from the dataset.

Image labels were obtained by expert annotation, into one of nine different habitat classes, as shown in Table 3.1. However, these labels contained an unacceptable amount of noise; partly due to labelling error, but mostly because of genuine ambiguity between fine-grained habitat classes. As a result, the image labels were consolidated into 5 habitat classes (Figure 3.2), characterised by keywords "sand", "screw shell rubble", "reef / sand interface", "reef", and "kelp" (*Ecklonia Radiata*). The sand class also contains some images of silt, which were only observed during a single dive ("waterfall 05").

The AUV dive data are shown in Figures 3.3 and 3.4, with the class labels obtained at each location overlaid on the gridded bathymetry data.

3.1.3 Co-located multimodal data

Matched multimodal data is obtained by extracting a 15×15 bathymetry patch centred at the AUV position corresponding to each image, as illustrated in Figure 3.5. Since the AUV position does not correspond exactly to the grid cell centre locations, the matching bathymetric patch values are obtained using linear interpolation in the grid. With a separation of 1.6m between grid points, each patch represents an area of 22.4m $\times$ 22.4m. It is important to stress that this area is much larger than the $2 - 3\,\mathrm{m}^2$ typically covered by an acquired image: due to the 1.6m spacing of the bathymetric grid, a bathymetric patch matching the footprint of the visual images would not capture much local structure.

The selected size of this region is based on two considerations: it has to be sufficiently large to capture enough texture in the bathymetry, and sufficiently small to avoid covering many different habitat classes. We note that the approach outlined in [5] uses multi-scale features up to a 50m $\times$ 50m area.

One potential concern that may affect multimodal matching is the presence of errors in the localisation of the AUV. However, the AUV navigation accuracy is comparable

Table 3.1 – Number of labels for each habitat class

Habitat Class	Number of images
Sand	3047
Coarse sand	4092
Sand & screw shell rubble	11901
Screw shell rubble	10728
Patch reef	5366
Reef-sand ecotone	3067
Low relief reef	10246
High relief reef	21681
Kelp	5028

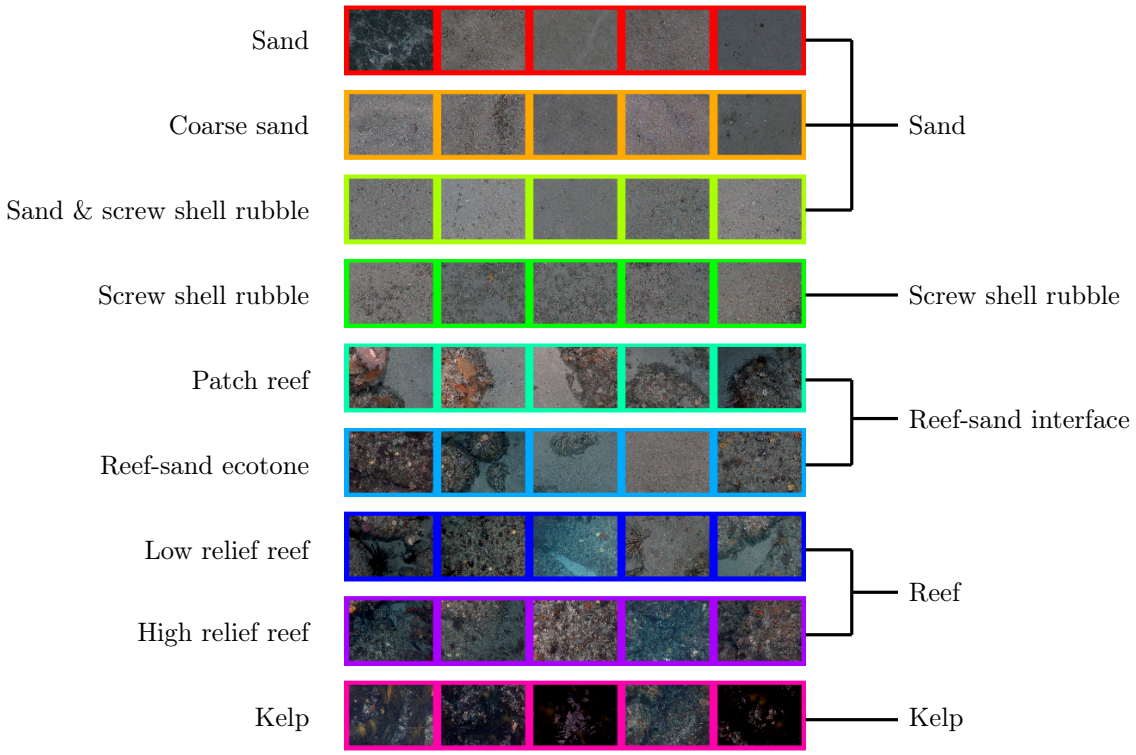
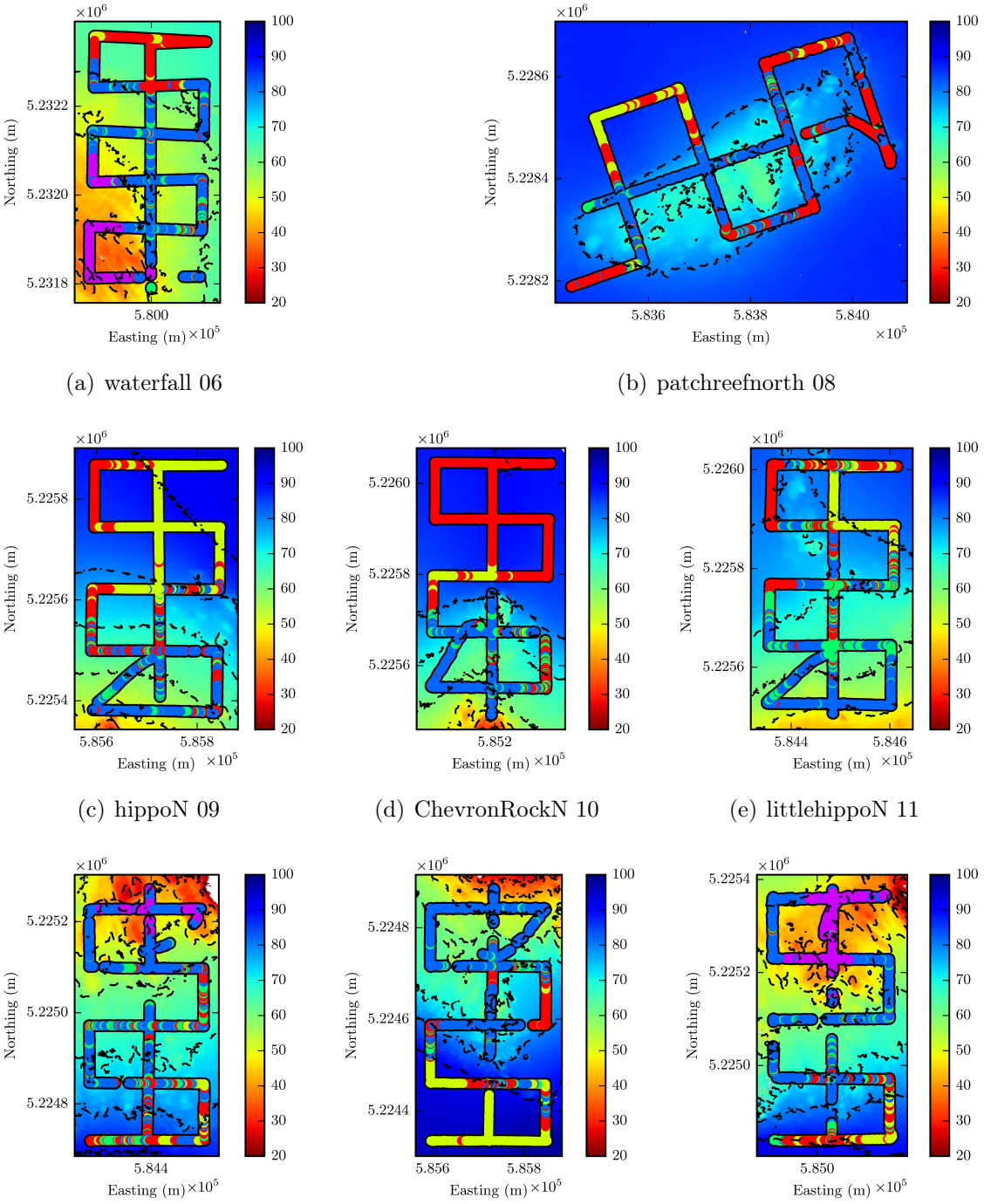


Figure 3.2 – The original label classes for the data, and the consolidated habitat classes. There is visual ambiguity between some habitat classes, as well as a small amount of labelling noise present. The sand class also contains some images of silt, an example of which is shown on the top left.



(a) waterfall 06

(b) patchreefnorth 08

(c) hippoN 09

(d) ChevronRockN 10

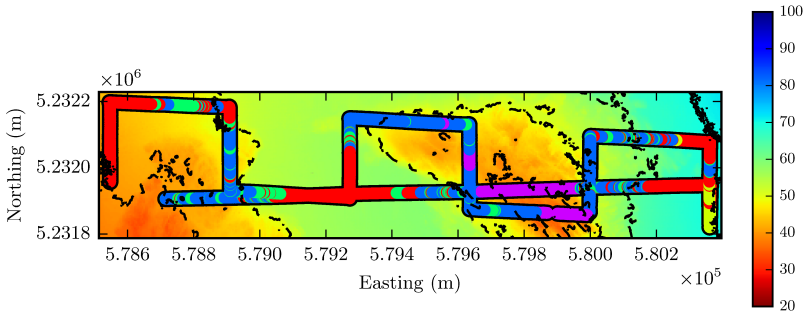
(e) littlehippoN 11

(f) littlehippoSE 12

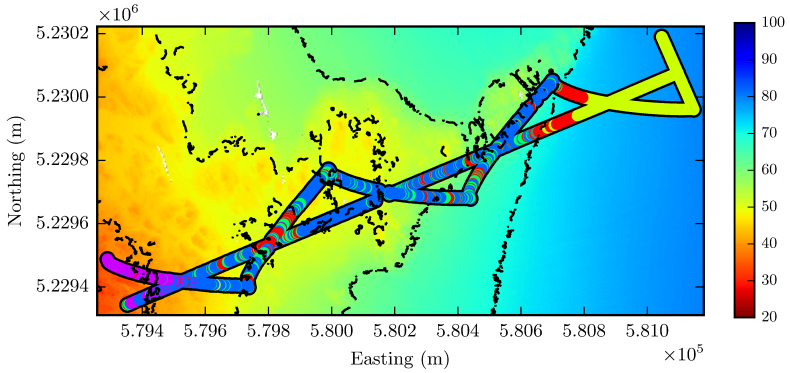
(g) hippoS 13

(h) ChevronRockS 14

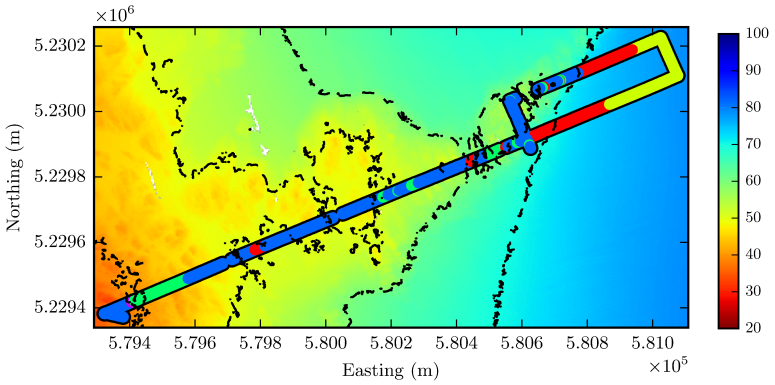
Figure 3.3 – The surveys performed in Southeastern Tasmania in 2008. The local bathymetry (coloured by depth) is overlaid by the AUV trajectory (coloured by class label). The colours for the bathymetry are indicated by the corresponding colourbars, while the class labels are sand (red), screw shell rubble (yellow), sand / reef interface (green), reef (blue), and kelp (purple).



(a) waterfall 05



(b) ohara 07



(c) ohara 20

Figure 3.4 – The surveys performed in Southeastern Tasmania in 2008. The local bathymetry (coloured by depth) is overlaid by the AUV trajectory (coloured by class label). The colours for the bathymetry are indicated by the corresponding colourbars, while the class labels are sand (red), screw shell rubble (yellow), sand / reef interface (green), reef (blue), and kelp (purple).

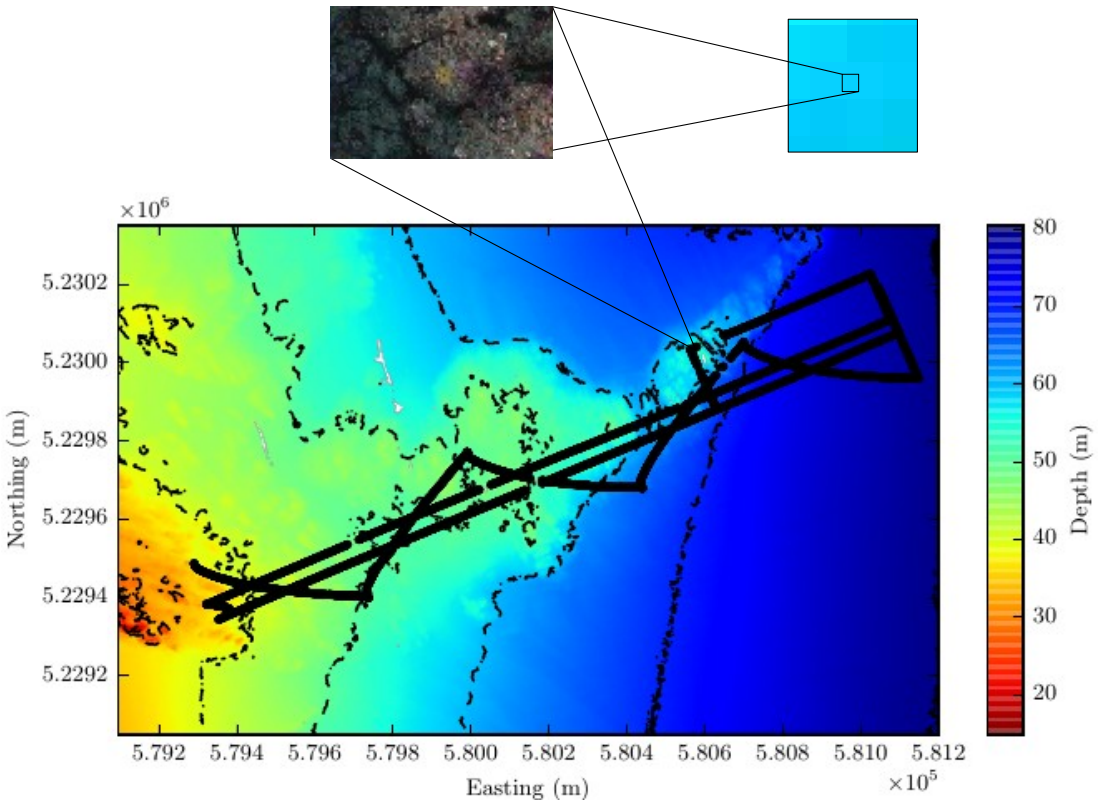


Figure 3.5 – An illustration showing how images from an AUV transect are matched to the corresponding bathymetry. At each image location along the AUV trajectory (shown in black), a corresponding 15×15 patch of gridded bathymetry is extracted. The patch covers a much larger area than the image footprint.

to the bathymetric grid spacing, and the habitats of interest typically vary at much larger scales. We therefore assume that any potential misregistration between the images and bathymetry as a result of localisation errors has a minimal effect on the relationship between the two modalities.

The final labelled multimodal dataset contains $75,427$ visual images, each matched with a bathymetric patch. Examples of matched images and local bathymetry are shown in Figure 3.6, grouped by habitat class label.

In all of the classification experiments, it is important to properly gauge the ability of the model to perform inference on unseen data. As a result, the multimodal dataset is divided randomly into a training set and a test set, both of equal size.

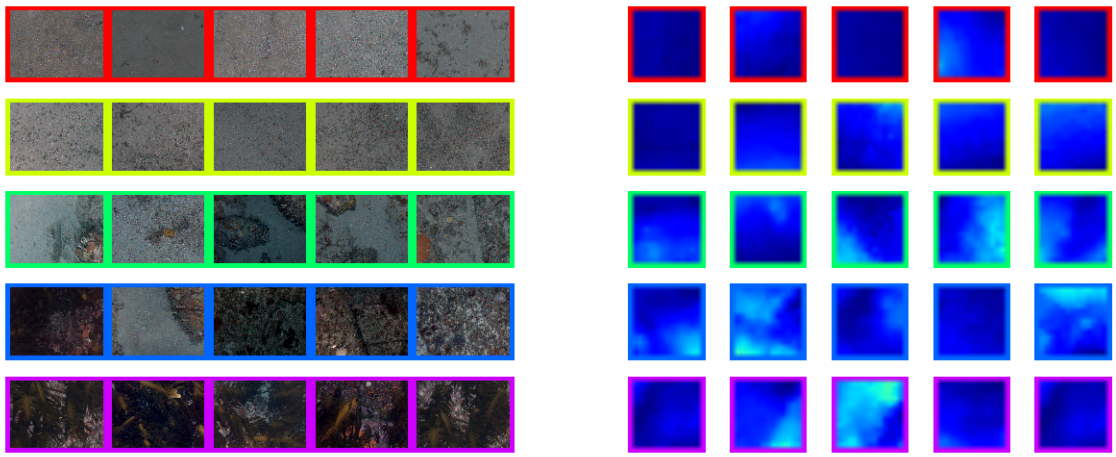


Figure 3.6 – Examples of the marine data corresponding to the different habitat classes (each row). Each image (left) is matched with its corresponding bathymetric patch (right). While the images typically have a footprint of approximately 1.2m × 1.5m, the bathymetric patches cover an area of 22.4m × 22.4m. The classes from top to bottom are sand (red), screw shell rubble (yellow), sand / reef interface (green), reef (blue), and kelp (purple).

3.1.4 Notation

The proposed algorithms utilise square patches of gridded bathymetry and AUV-based visual images. A bathymetry patch $\mathcal{B}$ can be considered as the sum of a mean ocean depth $\mathcal{B}_0 = \text{mean}(\mathcal{B})$, and a zero-mean patch capturing the local bathymetric variation (or "shape"), $\mathcal{B}_l = \mathcal{B} - \mathcal{B}_0$. The local variation is important in determining the habitat; for example, sandy regions are likely to exhibit smoother bathymetry gradients than reef habitats. Similarly, the depth is also significant, as, for example, kelp species prefer shallower water. However, since the depth has a much larger magnitude than the local variation, it is likely to dominate the feature representation if $\mathcal{B}$ is used directly. Put simply, if feature learning is performed on the raw patches, the model will primarily learn the depth, as it is the dominant factor of variation. This problem can be addressed by separating the bathymetry data into these two variables. For the remainder of this thesis, the mean ocean depth is refered to as $\mathcal{B}_0$, the zero-meaned local bathymetry patch as $\mathcal{B}_l$, and the visual input as $\mathcal{V}$.

3.2 Classification problem setup

One of the key aims of this thesis is to classify marine data into the class categories described in Section 3.1. This may involve large-scale benthic habitat mapping, in which only bathymetry is available, or it may involve classification of image data, or indeed, both modalities together.

While there are several complex and highly nonlinear models that could be used for the classification task, extensive analysis with different classifiers is outside the scope of this thesis. Instead, this thesis adopts the philosophy of unsupervised feature learning and deep learning, which suggests that application of feature learning techniques can significantly simplify the classification task into a linear separation problem. In particular, the use of feature learning techniques is usually equivalent to utilising a nonlinear classifier in the input space: the nonlinearity is simply absorbed into the feature learning stage rather than the classification stage. As a result, most deep learning works utilise a simple linear classifier for discriminative tasks [18, 55, 67–69, 73, 83].

For this thesis, a multinomial logistic regression classifier, also known as a *softmax* classifier, will be used for all experiments. Suppose we have a feature vector $\mathbf{x}$, and the label $\mathbf{y}$ is a multinomial random variable, taking on one of K different values: a one in the k^{th} dimension indicate the feature has been classified as class k, and a zero indicates otherwise. The softmax model computes a linear score function $f_k(\mathbf{x}) = \mathbf{w}_k \cdot \mathbf{x}$ for each class, where $\mathbf{w}_k$ is a vector of weights for the corresponding class. The predictive class probabilities are then proportional to the exponent of the score function values (Equation 3.1).

$$p\left(\mathbf{y}_k = 1 \mid \mathbf{x}\right) = \frac{e^{\,\mathbf{w}_k \cdot \mathbf{x}}}{\sum_{k=1}^{K} e^{\,\mathbf{w}_k \cdot \mathbf{x}}} \tag{3.1}$$

In practice, softmax classifiers are usually trained by SGD to maximise the cross-entropy between the true labels and predicted labels, and a regularisation term (the $\mathcal{L}_2$ norm of the weights) is included to prevent overfitting. The corresponding regularisation parameter can then be tuned to control the impact of this term.

For all of the classification tasks in this thesis, a three-fold cross-validation is performed on the training set to select the best out of a range of different regularisation parameters. The model is then trained on the entire training set using the optimal parameter value, and the results on the held-out test set are reported. This ensures that (a) the regularisation parameter is chosen in a principled manner, and (b) the reported classification accuracy is based on unseen data.

3.3 Bathymetric Feature Learning

The bathymetric features are split into two categories: the depth features (based on $\mathcal{B}_0$), and the local bathymetry features (based on $\mathcal{B}_l$). This section decribes the feature learning and encoding technique for each of these.

3.3.1 Local bathymetry $\mathcal{B}_l$

For the local bathymetry, a number of preprocessing steps are performed to ease the feature learning process. This is essential in order to learn good features, as corroborated by some previous research [17].

As a first step, the patches are individually normalised by dividing by their standard deviation: this is especially important for bathymetric data, since some patches have a very large depth variance, and will end up dominating the feature learning stage. To avoid amplifying noise in low contrast patches, the divisor is capped, based on the average standard deviation over the whole dataset. Note, however, that once the DAE has been trained, the *unnormalised* patches can be encoded using the model. The unnormalised patches are better for encoding, as the variance of each patch can actually provide information about the underlying habitat. Effectively, patches are only normalised to aid learning of the feature dictionary, and the raw patches are encoded.

The second preprocessing step is to perform whitening using PCA, which makes it easier to train a gradient-based model on the data [9]. PCA finds a linear projection

to convert the highly correlated input data into a space in which the dimensions are uncorrelated. Whitening involves scaling the resulting data by the inverse of the variance of each dimension, such that each dimension has unit variance (i.e. the covariance matrix is the identity). Finally, as the input data is often highly redundant, it is common to simultaneously perform dimensionality reduction in order to reduce the computational overhead during learning.

3.3.2 Depth $\mathcal{B}_0$

Typically in computer vision applications, the patches are zero-meaned prior to learning, and the resulting biases / constant values are discarded, since they represent changes in illumination or shade rather than genuine structure. However, this application is unique: the depth value subtracted from each patch is actually useful, but cannot be included directly as it will end up dominating the bathymetric feature representation.

Since the $\mathcal{B}_l$ features are the output of a DAE encoding (a sparse code in the interval $[0, 1]$), it is inadvisable to include the depth value directly, because it has a much larger variance and usually dominates the feature learning process. To address this, a modified 1-of-k encoding is employed for $\mathcal{B}_0$. This significantly aids the multimodal process by ensuring all feature dimensions have the same input range and similar statistics.

In this type of encoding, the observed depth range of $19 - 100\,\mathrm{m}$ is discretised into 82 equally spaced bins with an increment of 1m, and each $\mathcal{B}_0$ is encoded as a value of 1 for the corresponding depth bin. This type of "one-hot code" is used extensively in the literature [16, 18, 60]. One consideration is that input feature dimensions are considered independent for each autoencoder layer, so a standard one-hot encoding does not explicitly encode the continuous nature of the depth data, i.e. capture the fact that neighbouring bins are correlated. As such, a modified encoding is used, wherein neighbouring bins are encoded with a Gaussian-like falloff, with 0.8 in the immediately adjacent bins, 0.2 in the following bins, and 0 elsewhere. The encoding

values, and the width of this falloff are selected so that the sparsity of the encoded $\mathcal{B}_0$ features is approximately the same as that of the midlayer $\mathcal{B}_l$ features. The choice of the number of depth bins is a tradeoff between the ability to resolve fine-grained changes in depth and the amount of data required to train the model. With fewer depth bins, it becomes much more difficult to differentiate between kelp and reef habitats in shallow waters, as both classes fall into one bin. With greater depth resolution, we have found that the performance of the model suffers, most likely due to the fact that there are fewer training samples falling within each bin. The drawback of this encoding is that information is lost by discretising the continuous data into a set of depth bands, as different depth values that fall within the same bin are represented identically when encoded. To address this, the depth signal is linearly interpolated at the bin locations rather than actually discretised. That is, instead of directly using the one-hot code with Gaussian-like falloff as the depth signal, it is centred at the observed depth and interpolated at the depth bin locations.

This architecture is justified by considering the kinds of correlations that are likely to occur between these modalities. The ocean depth is unlikely to correlate with $\mathcal{B}_l$ patch pixels directly, but may be related to the first layer $\mathcal{B}_l$ features (local edge and gradient filters). For example, in the datasets used for this work, deeper areas often have smoother bathymetry gradients corresponding to sand habitats, while shallower reef regions exhibit localised 'blob-like' bathymetry.

3.3.3 Experiments

This section outlines the experiments on bathymetric feature learning. The local bathymetry patches are first normalised, and then whitened using PCA. The original 225-dimensional space is projected to 104 dimensions, to preserve 95% of the original variance. Feature learning is then performed by training a DAE with 1000 units on these normed, whitened values. Experimentation with different numbers of hidden units suggests that a dimensionality of 1000 represents a good compromise between classification accuracy and computational overhead. Since the data is real-valued, a

linear decoder is used for the autoencoder, as described in Chapter 2.3.

The DAE was written in Python using the pylearn2 library, and took approximately 24 hours to train on a NVIDIA GTX 590 GPU. The rugosity, slope and aspect features used as a baseline comparison were extracted using the MATLAB-based software libraries developed by [5].

3.3.3.1 Feature Learning

To understand what structure the autoencoder model has captured in the input data, the features learned by the model can be visualised as follows. As described in chapter 2.3, if we denote the weights of the model as $\mathbf{W}$ and the input data as $\mathbf{x}$, the activation function of the j^{th} hidden unit is given by $h_j = \text{sigm}\,(b_j + \sum_i w_{ij}x_i)$. To understand what this hidden unit is capturing, we try to find the input vector that maximises its activation function, subject to an $\mathcal{L}_2$ norm constraint to avoid trivial unbounded solutions. For this scenario, it can be shown that the solution for each hidden unit is simply the values of the weights themselves, scaled by the $\mathcal{L}_2$ norm over the corresponding weights vector. As a result, a simple way to visualise what each unit has learned is to plot its weights as a patch of input pixels [35].

If we visualise the weights learned for the bathymetric DAE, we obtain the features shown in Figure 3.7. Interestingly, the DAE learns edge and gradient filters similar to those obtained from natural image patches [17, 52].

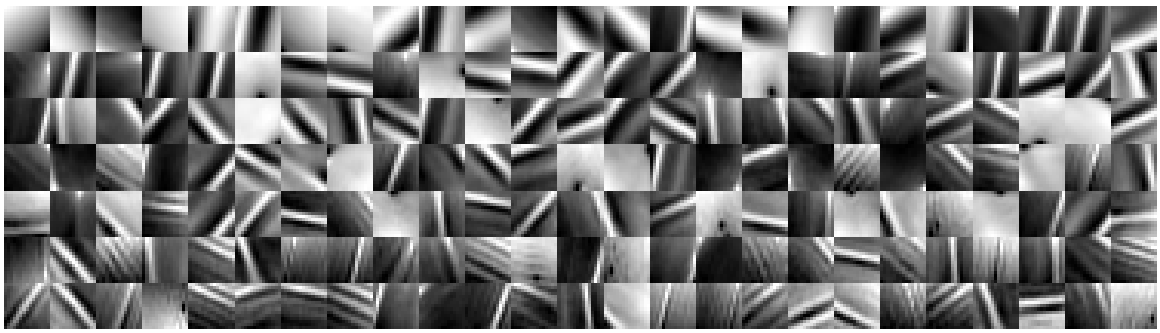


Figure 3.7 – A subset of the 1000 bases learned from 15×15 bathymetry patches, representing a $22.4 \times 22.4\,\text{m}^2$ area.

3.3.3.2 Analysis of traditional bathymetric features

This work is the first to utilise feature learning techniques on acoustic bathymetric data. As a result, additional analysis is presented to justify this approach, by comparing the learned features to hand-selected features typically used for bathymetric classification: multi-scale rugosity, slope, and aspect [5, 14, 100]. To compute these features, a given patch of bathymetry data is represented as a Delaunay triangulated surface mesh, and the plane of best fit is determined using PCA. The rugosity is then the ratio between the mesh surface area and the planar surface area, slope represents the angle between the plane of best fit and the horizontal plane, and aspect denotes the azimuthal direction of the surface slope [26]. These features are calculated on bathymetric patch sizes of 5×5, 9×9, 17×17, and 33×33, in order to correspond directly with the distance scales used by Bender et al. [5].

One way to quantify the success of bathymetric feature learning is to determine whether the hand-selected rugosity, slope, and aspect (RSA) features can be predicted by the learned representation. This provides an answer as to whether they are 'contained' within the learned features. Accordingly, Linear Least Squares is used to find the linear projection of the learned features that best matches the hand-selected features. For each RSA value, the Spearman rank coefficient (ρ) is calculated, indicating whether the relationship between the RSA feature and the projected feature is monotonic. The discriminative power of each of the RSA features can also be quantified by using them individually in the classification task. While many features may be most discriminative in conjunction with other feature dimensions, this measure still provides a rough measure of the value of individual features. These two metrics together provide a measure of (a) how well the learned features can predict each of the hand-selected features, and (b) the importance of the hand-selected features for classification tasks.

As the aspect variable is a representation of orientation, it is subject to angle wraparound, which means that, for example, π and $-\pi$ are identical. As a result, using the variable directly is not a good indicator for this exercise. Accordingly, we take the cosine and sine of the aspect variable at each scale, dividing it into a "northness" and "eastness",

and use both of these for the analysis. At each scale, the reported ρ value is the mean Spearman coefficient when predicting both northness and eastness, while the classification result is determined by classifying using both dimensions.

Table 3.2 – Spearman rank coefficient (ρ) when using learned features to predict rugosity, slope, and aspect features

	Scale			
Feature	5×5	9×9	17×17	33×33
Rugosity	0.751	0.850	0.902	0.877
Slope	0.782	0.783	0.765	0.720
Aspect	0.667	0.722	0.721	0.684

Table 3.3 – Classification accuracy (%) of rugosity, slope, and aspect features

	Scale			
Feature	5×5	9×9	17×17	33×33
Rugosity	52.43	54.49	57.45	56.87
Slope	55.62	56.05	55.23	53.68
Aspect	41.89	42.70	42.49	43.41

The Spearman coefficient values are shown in Table 3.2, and the classification accuracies of each of the features are shown in Table 3.3. From Table 3.2, we observe that the learned features are able to predict the rugosity, particularly at larger scales. Unsurprisingly, the most accurate prediction, with $\rho = 0.902$, is at the scale closest to the 15×15 patch size used for bathymetric feature learning. The results also suggest that the learned features contain a large amount of slope information, but do not capture the aspect features as well.

Looking at Table 3.3, we observe a similar relationship in terms of classification accuracy using each of the hand-selected features individually. The rugosity features have the largest discriminative power, particularly at the 17×17 scale, followed by slope, and then aspect. Interesting, the aspect features have very little discriminative power but are still somewhat captured by the learned features. This may be due to the fact that the learned bases are mostly edges or gradients, which inherently encode some information about orientation.

The results indicate there is a relationship between the discriminative power of rugosity, slope, and aspect features, and the ability to predict them from the learned features. This demonstrates the value of feature learning on bathymetric data. The algorithm is able to learn the structure of the data, and without supervision, extract the features and scales that tend to be the most discriminative in classification tasks.

3.3.3.3 Classification

Having compared the learned features to the more traditional rugosity, slope, and aspect (RSA) features, it is important to analyse their classification performance. As a result, we perform classification with five different feature combinations: RSA, $\mathcal{B}_l$, $\mathcal{B}_0$, RSA & $\mathcal{B}_0$, and $\mathcal{B}_l$ & $\mathcal{B}_0$. This is appropriate because the role of the learned features is to replace the hand-selected RSA features, and the depth value can be used in conjunction with either group. In each RSA scenario, the features over all scale are concatenated and used for classification.

The classification results are shown in Table 3.4. As a baseline, we also apply PCA directly to the raw and zero-meaned bathymetric patches (the latter only uses the local bathymetry, while the former also contains depth). As the results indicate, PCA is unable to extract useful local bathymetric features for classification, with an accuracy of 46%, but can extract the depth when applied to the raw patches, with 66% accuracy. However, the depth encoding proposed here ($\mathcal{B}_0$) still yields a higher classification accuracy of 67%.

Most importantly, it can be seen that the overall classification accuracy with the learned $\mathcal{B}_l$ features is 6% greater than with hand-selected RSA features, which supports the conclusions from the previous section. This accuracy is increased further to 72% by including the encoded depth features.

3.3.3.4 Habitat Mapping

By using the $\mathcal{B}_0$ & $\mathcal{B}_l$ features to perform classification on each location in the bathymetry grid, it is possible to perform large-scale benthic habitat mapping.

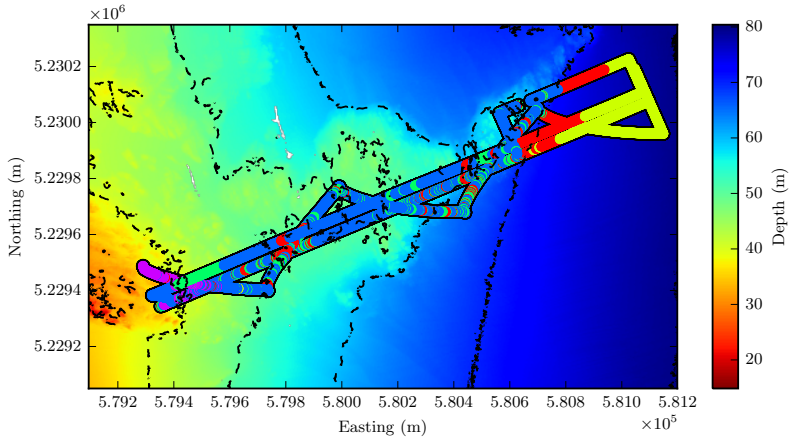
Table 3.4 – Classification accuracy of various bathymetric features

Data	Features	Classification accuracy
Local bathymetry patches	PCA	46.42%
	RSA	58.95%
	$\mathcal{B}_l$	64.60%
Depth only	$\mathcal{B}_0$	67.46%
Depth + local bathymetry	PCA	66.37%
	RSA & $\mathcal{B}_0$	68.86%
	$\mathcal{B}_l$ & $\mathcal{B}_0$	**72.57%**

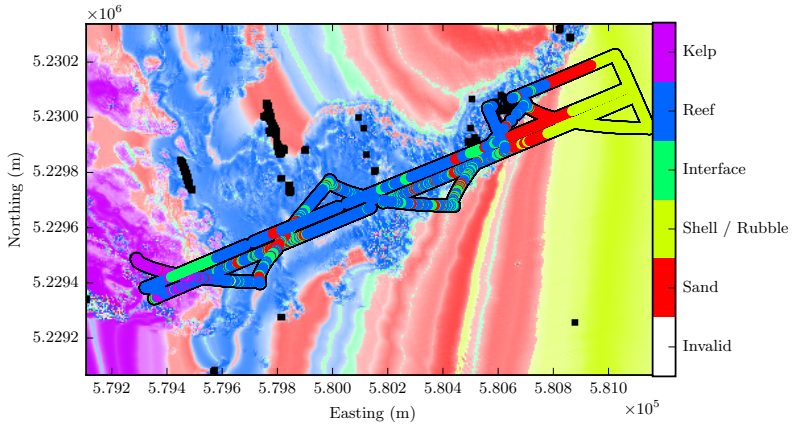
For this thesis, we analyse habitat mapping performance on a subset of the South-eastern Tasmania region, looking at a location known as O'Hara Bluff that is covered by two dives: "ohara 07" and "ohara 20". This area is of particular interest because it covers a large depth range (from $30 - 80$m), all five habitat classes, and contains a large proportion of the dataset in terms of images obtained. A diverse range of bathymetric features is exhibited over the region, from deep, flat-bottomed areas of screw shell rubble, rugose terrain within the bluff containing rocky reef outcrops, and dense kelp forests in shallow waters in the West. By analysing a smaller region, it is also easier to gauge whether the models match these expert predictions.

This experiment utilises the classifier trained in the previous section on the training dataset. To perform habitat mapping, bathymetric features are extracted for each point in the bathymetric grid over the O'Hara Bluff region, and fed into the classifier to obtain the habitat class probabilities and associated predictions.

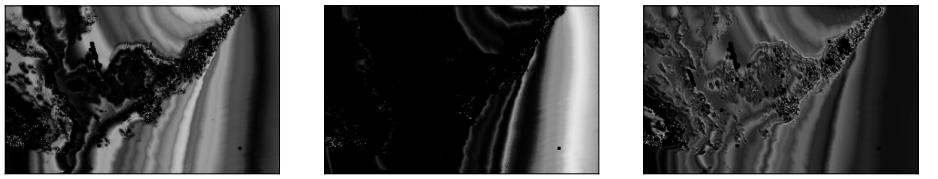
The habitat mapping results are shown in Figure 3.8. Figure 3.8(a) shows the bathymetry map for O'Hara with depth contours, and overlaid by the ground truth habitat labels assigned for each image along the AUV trajectory. The five class labels, from red through to purple, represent sand, screw shell rubble, reef / sand interface, reef, and kelp. Figure 3.8(b) shows the produced habitat map, where the colour represents the habitat class, and white regions represent areas over which bathymetry data was unavailable. The strength of the colour corresponds to the probability of the most likely class, such that the colour fades to white for a uniform distribution over classes. The individual class probability maps are shown in Figure 3.8(c).



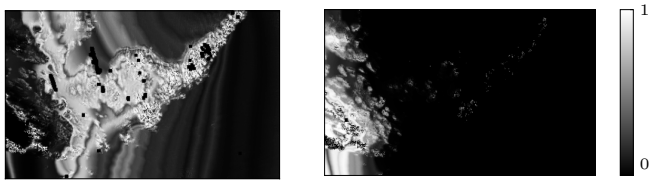
(a) Bathymetry map over the O'Hara Bluff region, overlaid with the AUV trajectory coloured by associated image labels.



(b) Habitat map using midlayer bathymetric features



(i) Sand (ii) Screw Shell Rubble (iii) Reef / Sand Interface



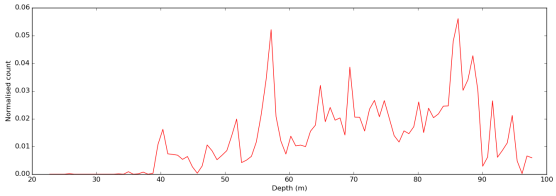
(iv) Reef (v) Kelp

(c) Individual class probability maps using multimodal layer features

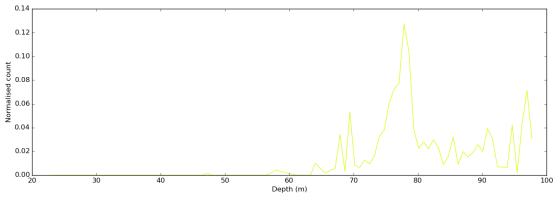
Figure 3.8 – Habitat mapping results for the O'Hara Bluff region using the bathymetric features $\mathcal{B}$. Each map is overlaid with the habitat labels corresponding to images taken during AUV transects in the area. The classes are sand (red), screw shell rubble (yellow), reef / sand interface (green), reef (blue), and kelp (purple). The habitat map fades to white in uncertain locations. These images are best viewed in colour.

In general, the habitat map is qualitatively similar to that produced by Bender et al. [5]. There appears to be a strong dependence on the depth, evidenced by the linear striations in the map. As expected, kelp is mostly found in the shallower waters towards the Southwest corner of the map, while screw shell rubble and sand are more likely to be observed in deeper waters towards the East, though they are distributed over a larger region. Similarly, reef is largely constrained to moderate depths, in highly rugose areas.

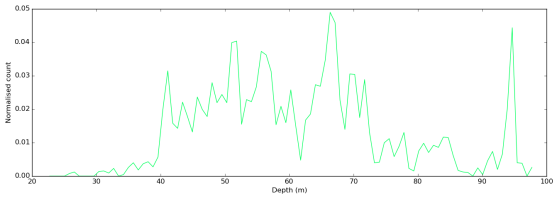
The individual class probability maps also provide some indications as to the distribution of each habitat class. The distributions of some classes are very strongly correlated with depth, with very low probability of kelp outside the shallow regions, and screw shell rubble restricted to deeper waters. In contrast, the sand distribution suggests it could be observed with nonzero probability over a large region. This is in agreement with the actual depth histograms shown in Figure 3.9, which indicate that the kelp and rubble classes have a very strong dependence on depth.



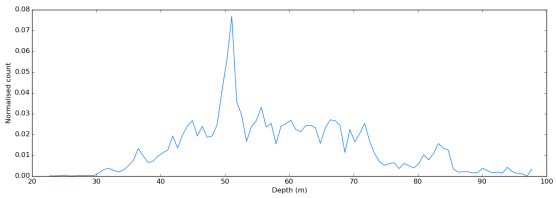
(a) Sand



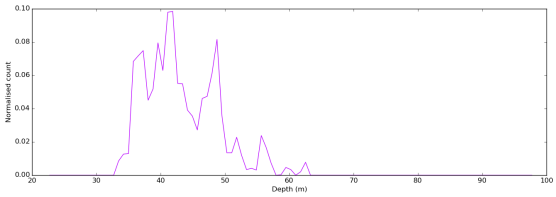
(b) Screw shell rubble



(c) Reef / sand interface



(d) Reef



(e) Kelp

Figure 3.9 – Depth histograms for each habitat class over the entire southeastern Tasmania region. The histogram plots are coloured according to the same scheme used for the five habitat classes throughout this chapter.

3.4 Visual Feature Learning

Having completed the bathymetric feature learning analysis, the next step is to consider feature learning from the AUV-borne visual image data. One key work in this area is that of Steinberg [89], who proposes a pipeline for the clustering of benthic imagery (and the Tasmania dataset in particular). The feature extraction pipeline is based on the Sparse Coding Spatial Pyramid Matching (ScSPM) technique [102], and numerous dictionary learning and sparse coding techniques are investigated.

Whereas deep networks can learn multiple layers of features [52], ScSPM is a single layer feature learner, followed by hierarchical pooling. In fact, similar approaches have been shown to do surprisingly well in the literature [10, 11, 50, 102], in some cases outperforming their deep counterparts. Nonetheless, there may be some benefit to utilising a convolutional network architecture to learn feature hierarchies, with, ideally, each layer representing a higher-order feature abstraction (i.e. from texture filters to object parts to whole objects). In this section, we describe the ScSPM-based approach, and compare its classification performance with various convolutional network architectures.

In practice, there are a variety of design choices with regards to the dictionary learning algorithm, sparse encoding technique, and various other steps in the ScSPM pipeline. Here, we only describe the procedure adopted by Steinberg [89], and direct the reader to [89] and [102] for a more in-depth analysis.

3.4.1 Sparse Coding Spatial Pyramid Matching

The ScSPM method is a feature learning and encoding technique that has been shown to produce features that are successful in image classification tasks. The algorithm has three main stages:

1. **Dictionary Learning**: A dictionary, or *codebook*, is learned from image patches.
2. **Sparse Encoding**: Image patches are encoded to a feature representation according to the learned dictionary.

3. **Spatial Pyramid Matching**: The encoded features are then pooled over a series of grids at different scales.

These are outlined in greater detail in the following sections.

3.4.1.1 Dictionary Learning

Given a large dataset of images, the first step is to extract a set of random sub-patches, and compute the SIFT descriptor vector for each one. Given this set of SIFT descriptor vectors $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N]$, the K-means clustering algorithm is used to learn a dictionary of cluster centres $\mathbf{D} = [\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_K]$, to describe the data, using the following objective function:

$$\min_{\mathbf{D},\mathbf{z}} \sum_{n=1}^{N} \sum_{k=1}^{K} \mathbf{1}(z_n = k) \, \|\mathbf{x}_n - \mathbf{d}_k\|_2^2 \tag{3.2}$$

Here, $\mathbf{z} = [z_1, z_2, \ldots, z_N]^\top$ represents the cluster assignments for each input vector, where $z_n \in \{1, 2, \ldots, K\}$, and $\mathbf{1}(\cdot)$ is the indicator function, which has a value of 1 if its argument is true and 0 if its argument is false.

After initialising the cluster centres randomly, the objective function is minimised by repeating two steps iteratively. The first step is the assignment step, where each data point is allocated to a cluster based on Euclidean distance to the centre:

$$z_n = \operatorname*{argmin}_k \|\mathbf{x}_n - \mathbf{d}_k\|_2^2 \tag{3.3}$$

The second step is the update step, where the cluster centre / dictionary element is updated to reflect the mean of all points assigned to it:

$$\mathbf{d}_k = \frac{\sum_{n=1}^{N} \mathbf{1}(z_n = k) \, \mathbf{x}_n}{\sum_{n=1}^{N} \mathbf{1}(z_n = k)} \tag{3.4}$$

With this procedure, the K-means algorithm models the input data as a set of spherical clusters, and the set of cluster centres forms the dictionary (or *codebook*) $\mathbf{D}$.

Algorithm 3.1: Orthogonal Matching Pursuit algorithm

1: $\mathbf{r} \Leftarrow \mathbf{x}$, $\mathcal{S} \Leftarrow \emptyset$
2: **for** $i = 1$ to T **do**
3: $\quad \hat{k} \Leftarrow \text{argmax}_k \left|\mathbf{r}^\top \mathbf{d}_k\right|$
4: $\quad \mathcal{S} \Leftarrow \mathcal{S} \cup \mathbf{d}_{\hat{k}}$
5: $\quad \mathbf{r} \Leftarrow \mathbf{x}_{\perp \text{span}(\mathcal{S})}$
6: **end for**

3.4.1.2 Sparse Encoding

The Sparse Encoding stage in this pipeline is performed using the Orthogonal Matching Pursuit (OMP) method (Algorithm 3.1), a greedy algorithm that sequentially finds the 'best' dictionary elements for a given input vector [70].

The algorithm maintains a set of selected dictionary elements $\mathcal{S}$, and a residual vector $\mathbf{r}$ (initialised to the input vector $\mathbf{x}$). At each step, the algorithm finds the dictionary element $\mathbf{d}$ with the largest correlation to the current residual, and adds this to $\mathcal{S}$. It then projects $\mathbf{x}$ to the span of the elements of $\mathcal{S}$, and computes the new residual $\mathbf{r}$ (i.e. the vector component of $\mathbf{x}$ that is orthogonal to the span of $\mathcal{S}$). This process is repeated until T dictionary elements are chosen.

3.4.1.3 Spatial Pyramid Matching

Given an input image, the Spatial Pyramid Matching process is as follows. First, a grid of overlapping patches is extracted from the input image. For each of these patches, the corresponding SIFT descriptor is extracted, and a sparse encoding is obtained using the OMP technique. This results in a K-dimensional sparse code for each patch location in the image grid.

The spatial pyramid itself consists of a number of pooling layers, each over a successively larger region of the image. In each spatial pyramid layer, the image is divided into a uniform n-by-n grid, and the sparse codes are max-pooled over each grid cell. The pooled codes from all spatial pyramid layers are then concatenated into a single

feature vector. Typical values for the spatial pyramid layers are $n = \{4, 2, 1\}$, which means that the final feature vector has dimension $(4^2 + 2^2 + 1^2) \times K = 21K$.

As a result, the spatial pyramid matching stage effectively summarises the image content in terms of the presence of dictionary elements over different scales.

3.4.1.4 Additional Processing

In his work, Steinberg [89] proposes using a dictionary of size $K = 1024$, which means that the ScSPM feature vector is 21504-dimensional. Such high-dimensional data can be prohibitively expensive for classification tasks. As a result, an additional dimensionality reduction stage is also proposed, reducing the data to 3000 dimensions using Random Projections [3]. Steinberg [89] demonstrates empirically that this data compression is achieved with minimal loss in classification performance.

Finally, for this thesis, an additional DAE layer is trained on the 3000 dimensional features, to obtain a lower dimensional sparse code. With this step, the visual features now lie in the interval $[0, 1]$, and a sparsity cost during training can encourage the sparsity of the hidden units to be similar to that of the bathymetric layer. By ensuring that the top-level visual features have a similar statistical structure to the top-level bathymetric features, in terms of output range and sparsity, it is much easier to capture the relationship between the two.

3.4.1.5 Discussion

The proposed ScSPM-based pipeline has a number of similarities with the CNN model described in chapter 2.4. It involves learning a set of filters, each with a small receptive field, which looks at nearby pixels rather than the entire image. It involves a convolution of each of these filters over the entire image, resulting in a set of feature maps, each representing the response of a particular filter over the image. And finally, it also involves a number of pooling layers to reduce the resolution of the feature maps and suppress non-maximal activations.

As a result, the ScSPM model can be considered a form of CNN, consisting of a single feature learning layer, with multiple pooling layers. The key difference, then, is the higher-order feature learning layers that are present in a CNN. As such, it is prudent to also apply CNN models to this problem, to gauge the benefits of the high-level feature layers. This will be discussed in the following section.

3.4.2 Convolutional Neural Networks

Given that CNNs achieve state-of-the-art performance in many vision-based classification tasks [48, 52, 83], we also apply a number of CNN architectures to the image data as a benchmark.

Due to the very large memory requirements of CNNs on high-resolution images, each image is first downsampled by a factor of 4 to a size of 340×256. By visual observation, the downsampled images do not lose any important structure as compared to the original images.

All of the networks contain three convolutional and pooling layers, followed by a single fully connected layer. The model parameters are shown in Table 3.5. The parameters for the k^{th} layer are denoted by a subscript of k. The receptive field size is given by n_{w_k}, and the number of hidden units / feature maps by n_{h_k}. The stride parameter n_{s_k} refers to the number of pixels between adjacent applications of the filter in the convolutional layer. Pooling is performed over non-overlapping regions with n_{p_k}.

Some of the parameters are kept constant across all of the networks. As the feature maps of the first convolutional layer are quite large, it is common to specify a small number of hidden units. As such, the first convolutional layer contains only 25 hidden units. In fact, Krizhevsky et al. [48] recommend adopting a stride of 4 for larger images, and their 224×224 images are smaller than those used here. The number of hidden units in the final layer is set at 1000, in order to match the dimensionality of the ScSPM-based features. The final layer is fully connected, and in each case, the receptive field for this layer is a 2×3 feature map covering the entire image.

Table 3.5 – The parameters for the convolutional neural network models applied to visual classification. n_{w_i}, n_{h_i}, and n_{s_i} refer to the receptive field size, number of filters, and stride length of the i^{th} convolutional layer, while n_{p_i} is the pooling size in the i^{th} pooling layer.

		CNN 1	CNN 2	CNN 3	CNN 4
Layer 1	n_{h_1}	25	25	25	25
	n_{w_1}	9	9	8	5
	n_{s_1}	4	4	4	4
	n_{p_1}	2	2	4	2
Layer 2	n_{h_2}	100	50	100	100
	n_{w_2}	7	7	4	5
	n_{s_2}	2	2	2	2
	n_{p_2}	2	2	2	2
Layer 3	n_{h_3}	500	100	500	500
	n_{w_3}	4	4	3	5
	n_{s_3}	1	1	1	1
	n_{p_3}	2	2	1	2
Layer 4	n_{h_4}	1000	1000	1000	1000
	n_{w_4}	2×3	2×3	2×3	2×3

Many of the parameters are varied for each network, in order to best represent the different types of convolutional networks. The first network, CNN 1, represents a more conventional network, similar to that in [48]: for higher layers, the receptive field size and stride decreases, the number of hidden units increases, and the pooling ratio is kept constant at 2. The second network utilises fewer hidden units in the lower layers, quantifying the impact of the dimensionality on the model. In CNN 3, the pooling ratio is increased to 4 in the first layer and reduced to 1 after the third convolutional layer. This investigates whether the lower layer feature maps can be reduced in size by pooling, and determines how much information is discarded in the process. Finally, CNN 4 represents a model in which the receptive field sizes are kept constant over all of the layers.

As is the standard in the literature, rectified linear (ReLU) units are used for each layer in the network, and dropout is applied to the fully-connected layer. The output layer is the standard softmax classifier, and the networks are trained by backpropagation,

using SGD.

3.4.3 Experiments

This section describes the experiments for visual feature learning. The visual dictionary used for ScSPM is identical to [89], and is learned externally on a natural images dataset. Steinberg [89] demonstrated that this was equally as effective as training the dictionary on the marine dataset, with the added benefit of not having to retrain the model for subsequent marine dive campaigns.

The DAE layer was trained with 25% masking noise and consists of 1000 hidden units, to match the bathymetric feature learning in Section 3.3. Again, this provided the best compromise between dimensionality and accuracy.

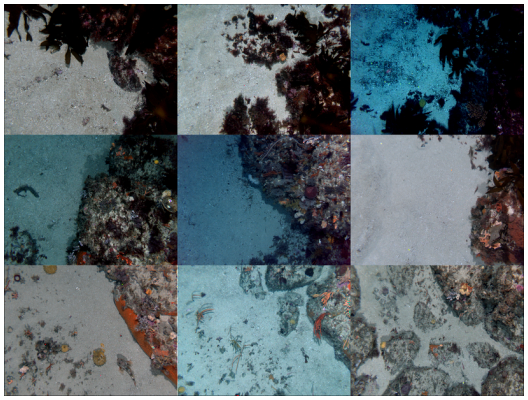
The DAE was written in Python using the pylearn2 library, and took approximately 34 hours to train on a NVIDIA GTX 590 GPU. The CNNs were also developed in pylearn2, and took approximately 9-10 days to train on the same GPU. The ScSPM pipeline used the software library developed by Steinberg [89] along with the pretrained dictionary from that work.

3.4.3.1 Learned features

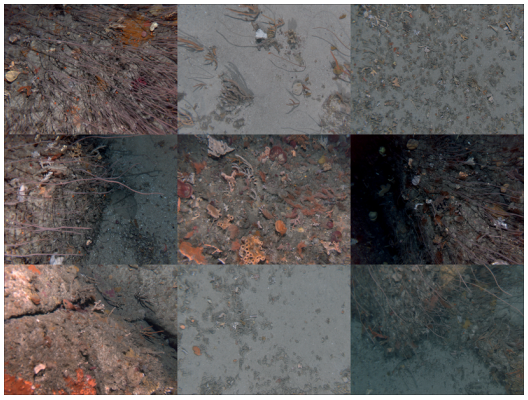
Whereas the first layer features of a neural network can be visualised by directly plotting the weights in input pixel space, no such simple technique exists for higher layers. In particular, when several pooling layers are used, it is not straightforward to visualise each learned feature. Previous research has highlighted various techniques for visualising high-level features, either by solving an optimisation problem to maximally activate each neuron [23], inverting the internal representations with a 'deconvolutional network' [20, 104], or plotting a saliency map for each object class to indicate class-critical regions of an image [81].

To understand the features that have been learned by the ScSPM pipeline, we instead adopt the computationally simpler approach of plotting the test set images

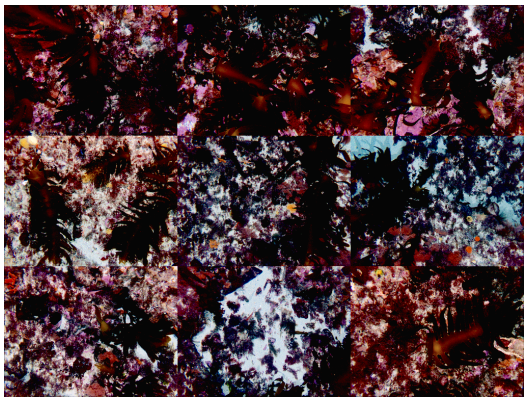
(a) Sand



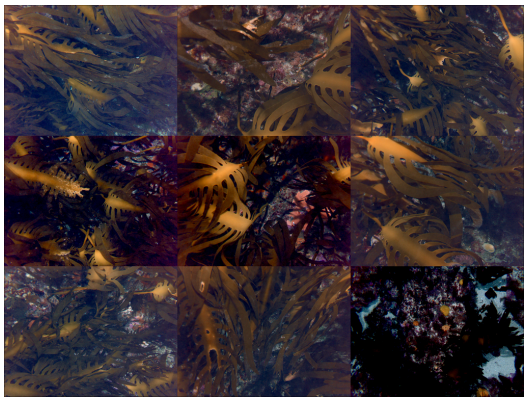
(b) Sand / reef interface, changing horizontally



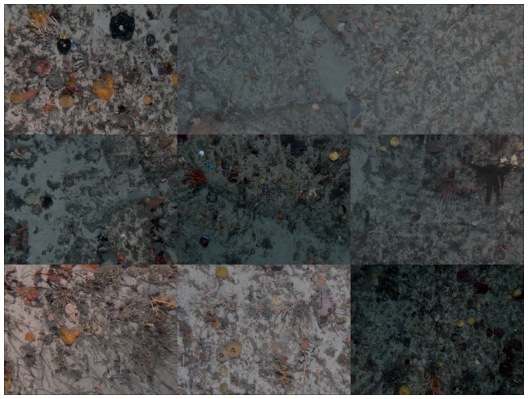
(c) Sponge



(d) Sparse kelp cover



(e) Dense kelp cover



(f) Patchy reef structure

Figure 3.10 – Visualisation of a small number of the learned image features, in terms of the top 9 input images that maximally activate each feature dimension. Accompanying each of the image groups is a description of the visual structure common to the images within each group. The features capture a variety of factors of variation in the data.

that maximally activate each feature dimension [23]. To achieve this, the entire test set is encoded according to the learned features, and for each dimension the top 9 input images are selected, such that they maximally activate the corresponding feature value. This provides some indication of the visual structure and content to which each learned feature is sensitive. Given that there are 1000 feature dimensions, the top 9 activations are shown in Figure 3.10 for only a small selection of learned features.

Figure 3.10 indicates that each of the learned features captures one aspect of variation in the data. Some of the features are specific to a particular habitat class (such as sand or kelp), while others capture additional structure, such as the orientation of reef / sand interface, or the presence of a particular species of sponge.

Thus, the features cover many different factors of variation in the data. Some act as class-based features, and are particularly useful for the classification task, while others capture textural or content information that may not be as useful. This is appropriate for this application: while classification is a key task of interest, it is also desirable to capture the additional factors of variation, and avoiding restricting the model by the habitat labels that are present. This ensures that the model is still useful for other tasks and is even applicable for different habitat categories (for example, for fine-grained species classification).

3.4.3.2 Classification

We can now compare the classification performance of the proposed ScSPM-based pipeline with the CNN architectures. For the ScSPM case, the top layer features are used to train a softmax classifier using the same training and validation procedure as with the bathymetric features. The CNNs are directly trained to minimise classification error.

The classification results are shown in Table 3.6. The first CNN model offers a small improvement over the ScSPM features, but the other CNNs perform more poorly. Nonetheless, the classification accuracies are very similar across all five models (within

Table 3.6 – Classification accuracy of visual features

	Classification accuracy
ScSPM	79.98 %
CNN 1	80.76 %
CNN 2	79.65 %
CNN 3	77.88 %
CNN 4	78.23 %

3%). This suggests that the main benefit comes from the first layer of feature learning, and higher layers have minimal effect.

One explanation for this behaviour is that the marine images have a very different structure compared with the image datasets to which CNNs are usually applied (objects, urban scenes, outdoor environments, etc). These datasets can usually be decomposed at different scales into objects, object parts, and lower level structures like edges. In marine images, this hierarchy does not appear as strongly: many of the images are sand, and even the reef or sand / reef interface images often do not contain objects at a larger scale in the image. As such, the image can be well described as a collection of low-level edges and textures, as characterised by the ScSPM approach.

For this reason, the ScSPM-based pipeline will be used for the visual feature learning layers in future chapters. While the first CNN was able to achieve a slight improvement in classification performance, its computational load in terms of training was far greater. Further, the dictionary for ScSPM can be learned on an external dataset and has been shown to generalise well when applied to different marine environments [89]. In contrast, a CNN-based approach would likely require training a separate model from scratch on a new marine dataset.

Ultimately, a more comprehensive study of these techniques would yield greater insight as to their efficacy on marine data. For the purposes of this thesis, in which the focus is on multimodal learning, the ScSPM-based approach will be adopted.

3.5 Summary

This chapter introduced the marine dataset used in this thesis, and proposed algorithms to learn features from visual and bathymetric data. The visual images dataset was obtained by the AUV *Sirius* over the course of 11 dives in Southeastern Tasmania in 2008, while the bathymetry data was obtained via Geoscience Australia's bathymetric grids [84].

Feature learning models were introduced for both bathymetry and visual data. The learned bathymetric features were compared with the traditional hand-picked features of rugosity, slope, and aspect, in terms of classification performance, and were found to be superior. Habitat mapping results were also presented using these features. The visual feature learning pipeline was compared with convolutional neural networks, which have demonstrated state-of-the-art performance in many visual classification tasks, and was shown to be competitive.

The following chapters will incrementally build on the proposed single-modality feature learning models, in order to model the relationship between the two modalities.

Chapter 4

Multimodal learning from visual and bathymetric features

This chapter outlines a multi-layer architecture to perform multimodal learning from visual images and bathymetric data. The model is trained in an unsupervised fashion, and can be used to perform both discriminative and generative tasks. The results presented in this chapter include classification of different modality combinations, and multimodal inference tasks such as sampling one modality from the other. Section 4.1 describes the multimodal model, and Section 4.2 describes the inference procedures for the model. Section 4.3 then presents the experimental results with this model for both classification and sampling tasks.

4.1 Model description

The proposed architecture learns multimodal correlations using a multi-layer hierarchy, in a similar fashion to the previous work outlined in 2.5. As shown in Figure 4.1, the features for $\mathcal{B}_0$, $\mathcal{B}_l$, and $\mathcal{V}$ (as detailed in Chapter 3) are concatenated in the *mid-layer*. A DAE *shared layer* is then learned using the midlayer as input, which learns the correlations between the features of each modality. The stochastic corruption applied during training is *masking noise*, in which input dimensions are randomly set

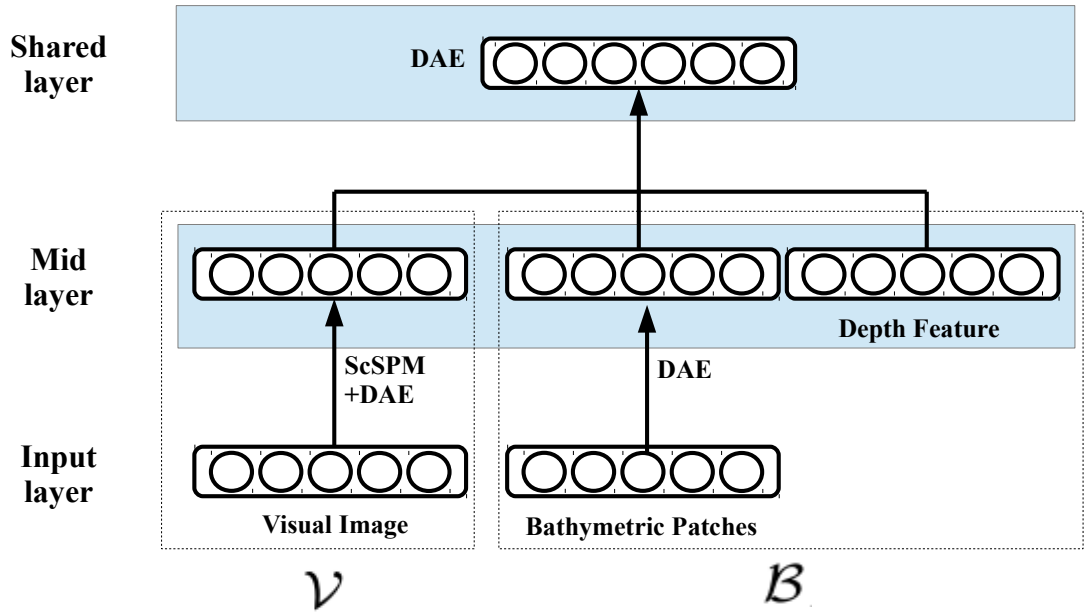


Figure 4.1 – The proposed model for multimodal learning. The ScSPM+DAE visual features, DAE bathymetric features, and encoded depth features are concatenated at the mid-layer, and the multimodal DAE is learned on top.

to zero, and the model is trained to reconstruct the uncorrupted input. As with the midlayer feature learners, the multimodal DAE was tested with different numbers of hidden units, and it was found that the choice of 2000 units exhibited the best compromise between performance and computational load.

The choice of DAE for this network is justified for a number of reasons. The first is that the use of masking noise encourages the model to be robust to missing input data. This is particularly important for this application, because the model needs to be able to perform inference when only one of the modalities is available. For example, it should be able to classify in-situ image data, even if there is no associated bathymetry. More commonly, for the benthic habitat mapping case, large-scale gridded bathymetry data is available, but visual images are only available over a small fraction of the region. The second reason is that the probabilistic properties of the DAE (described in

Section 2.3) facilitate generative tasks, such as the sampling of one modality given the other. Such inference tasks can help to understand the sorts of key correlations that exist within the data, in terms of which features from each modality are likely to co-occur. The third reason is that its objective function makes it easier to train. Whereas the RBM approximates the maximum likelihood gradients through the Contrastive Divergence (CD) algorithm, autoencoder models have a simpler reconstructive error objective that ease the learning process. Given these benefits, the DAE is a better option than an RBM for the shared layer.

As mentioned previously, the shared layer learns the correlations between visual and bathymetric features, which means that the resulting feature space (the hidden unit representation) captures features from both modalities. Another way to interpret this is that it performs a nonlinear projection of the data into a new feature space, and learns this projection in such a way that the data from both modalities can be well reconstructed. This means that, even when only one modality is available, the single modality data is embedded in a feature space that facilitates reconstruction of both modalities. One would expect that this embedding would lead to improved classification performance over the midlayer representation, especially when one modality is unavailable.

4.2 Inference

4.2.1 Classification and habitat mapping

For the DAEs utilised in single-modality learning in Chapter 3, the encoding procedure was simply to obtain the hidden unit representation of the input data. In the multimodal scenario, it is important to obtain features for any combination of input modalities.

To perform benthic classification with both the bathymetry data ($\mathcal{B}_0$ and $\mathcal{B}_l$), and the visual data $\mathcal{V}$, it is possible to do a forward-pass up the network to obtain the shared layer feature representation, and pass these "multimodal encoded" features

into the linear classifier. When modalities are missing, a simple encoding technique is to set the missing input dimensions to zero in the midlayer before performing the multimodal encoding. Given that the DAE learns features robust to masking noise, one would expect that this procedure will yield better results than if we were to perform classification on the midlayer features directly [69].

4.2.2 Prediction and sampling

By using a shared layer that covers the features of both modalities, the model is able to perform multimodal prediction and sampling tasks. One benefit of the DAE is that it can produce plausible samples from the data-generating distribution, in a similar fashion to an RBM.

However, contrary to RBMs, DAEs lack a model of the marginal distribution of the hidden layer [94] and cannot generate samples from an arbitrary hidden layer representation. Vincent et al. [94] proposes that this marginal distribution be modelled as an empirical distribution, comprised of the set of hidden codes obtained by encoding the training vectors.

Thus, for a single layer DAE, a sample can be obtained as follows. First, a training sample is encoded to obtain the hidden layer representation. Then, Bernoulli sampling is performed, where the activation value of each unit acts as the probability of the unit turning on (as with an RBM), resulting in a binary code. Finally, deterministic decoding yields a new input sample.

For a multi-layer network, a similar procedure is adopted, comprising of a deterministic bottom-up pass, followed by alternating Bernoulli sampling and deterministic decoding. In other words, a training sample is encoded according to the shared layer, Bernoulli sampling is performed, the reconstructions for the layer below are obtained, and the process is repeated until the input reconstructions are obtained. Note that since the ScSPM approach cannot perform top-down decoding (due to the pooling layers), it is not possible to sample visual features below the image level. This is usually the case for approaches that perform pooling (including CNNs), since data is

lost in the downsampling process. However, this is not restrictive for the application in this thesis.

In the scenario where modalities are missing, they can be either predicted (deterministically) or sampled (stochastically), by clamping the known inputs at their observed values [69, 85]. Here, the inference procedure is identical to the full data case, except that the known inputs are kept at their observed values.

For example, if the task was to sample the bathymetric features associated with a given input image, the process would be as follows. First, the midlayer features would be obtained, with zeros for the missing bathymetric dimensions, and used to compute the shared layer representation. Then, the stochastic top-to-bottom pass from the aforementioned sampling process would be applied in order to obtain a sample of the bathymetric input data, but the visual inputs would remain unchanged. This process can then be repeated to obtain several conditional samples of bathymetric data given visual features. Typically, the process is repeated a number of times before the resulting sample is kept, to ensure that successive samples are less correlated.

If the task is to obtain a conditional expectation / prediction rather than a sample, then the above procedure is repeated without the Bernoulli sampling steps. On a single iteration, the model effectively uses the known input data to obtain a deterministic reconstruction of the missing inputs. By iterating several times (without stochastic sampling), the model converges on the conditional expectation of missing inputs given known inputs.

4.3 Experiments

This section outlines the experiments on classification and sampling using the multimodal model.

For the midlayer features, the preprocessing steps and midlayer architecture are identical to those described in Chapter 4. For the shared layer, 2000 hidden units are used: this is based on experimentation with a number of hidden layer sizes, and the

selected number exhibited good accuracy with low computational load. The layer is trained as a DAE with 50% masking noise, such that each dimension is set to zero or retained with equal likelihood. This ensures the layer is robust to missing inputs, which is necessary for this application, as either modality may be unavailable.

The DAE was written in Python using the pylearn2 library, and took approximately 2 days to train on a NVIDIA GTX 590 GPU.

4.3.1 Classification

In this section, the classification performance of the features extracted through multimodal learning is compared with that of the midlayer features, as described in Chapter 3. The classification setup and training / test split are identical to Chapter 3.

As previously discussed, there are various modality scenarios that may occur at classification: either bathymetry ($\mathcal{B}$) or visual features ($\mathcal{V}$) may be available on their own, or both modalities may be available ($\mathcal{B}$ and $\mathcal{V}$). The goal of this analysis is then to determine whether performing multimodal learning beforehand is beneficial in terms of classification performance, even if one modality is unavailable when it comes to classification / inference time.

To analyse this, the multimodal model was trained on the entire training set, and the shared layer representation was obtained (the "multimodal encoding"). This was repeated for each modality scenario, with missing dimensions set to zero in the midlayer, such that each scenario had a separate set of multimodal encoded features. The classification performance of these features was then compared with the midlayer features.

The classification performance is shown in Table 4.1, for each scenario. To gauge the relative significance of the depth feature $\mathcal{B}_0$ and the local bathymetry $\mathcal{B}_l$, they have been considered as separate modalities for this analysis, even though they are both extracted from the bathymetry data.

Table 4.1 – Classification performance for various input modalities, reported as % accuracy. The highlighted case is the benthic habitat mapping scenario.

Modalities used	Midlayer encoding	Multimodal layer encoding
$\mathcal{B}_0$ only	67.46	67.46
$\mathcal{B}_l$ only	64.60	70.20
$\boldsymbol{\mathcal{B}}$ **($\boldsymbol{\mathcal{B}_0}$ and $\boldsymbol{\mathcal{B}_l}$)**	**72.57**	**81.23**
$\mathcal{V}$	79.98	80.71
$\mathcal{B}_0$ and $\mathcal{V}$	82.11	84.44
$\mathcal{B}_l$ and $\mathcal{V}$	81.24	84.92
$\mathcal{B}$ and $\mathcal{V}$	83.05	87.43

Looking at the midlayer features, we observe that results are significantly better when visual data is present. This is to be expected, given that the habitat classes are much easier to disambiguate from the visual images than from the bathymetry data. When comparing the multimodal encoding with the midlayer encoding, we observe an improvement in performance for all modality scenarios, except for the depth feature $\mathcal{B}_0$, with which the performance does not change. This may be due to the fact that it only occupies 82 dimensions of the 2082-dimensional midlayer; as a result, it does not have enough expressive power to harness the benefits of multimodal learning on its own.

With only visual data available, the performance with multimodal encoding is only a marginal improvement over the midlayer encoding. This may indicate that the visual features are already fairly precise for this task, and gain little information by encoding their relationship with bathymetric features.

The most important result is the benthic habitat mapping case $\mathcal{B}$, where the multimodal approach yields a 9% improvement in accuracy. This result suggests that the discriminative power of bathymetric data is significantly improved by transformation to a feature space which encodes correlations with visual imagery.

In general, the results are consistent with the analysis performed by Ngiam et al. [69] for audio and video data, demonstrating that having both modalities present at feature learning time improves the quality of the features learned for each modality.

4.3.2 Precision and recall analysis

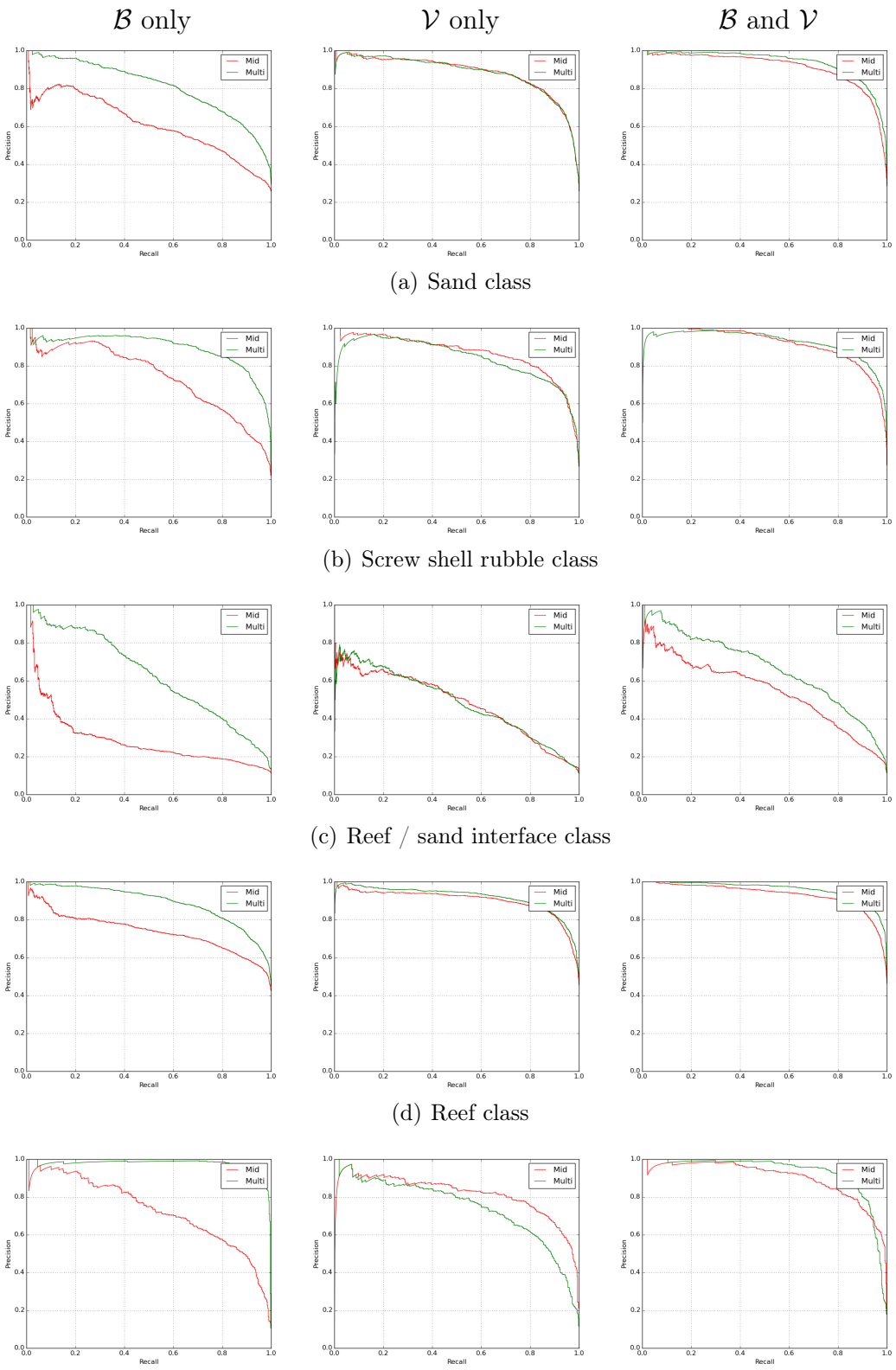
While the accuracy provides some information about classification performance, it often does not paint the full picture. To better understand the benefits and shortcomings of the multimodal learning approach, this section analyses the precision and recall of the two feature encodings.

Since precision and recall are binary classification measures, they cannot directly be applied to the multi-class problem. While it is possible to produce a 'micro-averaged' precision-recall curve that summarises the multi-class performance, a better approach, for the purposes of this thesis, is to compute the precision-recall curve for each class individually. This provides more information on how each classifier performs with respect to individual classes.

The precision-recall curves are shown in Figure 4.2. Each row refers to a separate class, while the plots within each row each refer to a separate modality scenario (ie. which modalities are available): from left to right, they are the $\mathcal{B}$ only, $\mathcal{V}$ only, and $\mathcal{B}$ and $\mathcal{V}$ scenarios. Within each plot, the classifier for the midlayer feature encoding (red) is compared with the classifier for the shared layer feature encoding (green). As such, each plot illustrates the effect of performing multimodal learning, for a particular class, for a particular modality scenario.

From the plots in the left hand column, it can be observed that the multimodal encoding has the greatest effect when only bathymetry data is available. This is consistent with the classification results, which demonstrate a 9% increase in accuracy for this scenario. In contrast, there is a lesser effect when both modalities are available (right hand column), and minimal change when only visual data is available (middle column). Looking at the different classes, we can observe that both models struggle the most with the reef / sand interface class. This is most likely due to the fact that the interface class is effectively a combination of reef and sand: consequently, both visual and bathymetric features may be quite similar to either of these classes, leading to ambiguity in the class labels.

For the bathymetry-only case, multimodal encoding appears to make the biggest



(a) Sand class

(b) Screw shell rubble class

(c) Reef / sand interface class

(d) Reef class

(e) Kelp class

Figure 4.2 – Precision-recall curves for each habitat class, for each modality scenario. In each case, the left hand plot is for the bathymetry only ($\mathcal{B}$) scenario, the centre plot is for the visual only ($\mathcal{V}$) scenario, and the right hand plot is for both modalities ($\mathcal{B}$ and $\mathcal{V}$).

difference for the classes with less representation in the dataset: kelp and reef / sand interface. However, there is also a sizeable improvement for the other three classes. For the case with both modalities, multimodal encoding offers a small improvement across the board, with the largest change for the interface class. This is an interesting result, as the interface class is particularly difficult to characterise, especially with bathymetric data, which locally can appear very similar to reef or sand classes. In fact, this result appears to indicate that the *cooccurrences* of visual and bathymetric features (i.e. the cross-modality correlations) are the most important in identifying reef / sand interface.

For the visual-only case, most of the plots indicate little change in performance, but interestingly, the result for kelp is poorer with multimodal encoding. This highlights one caveat in the learning process: while multimodal learning improves performance on average for all modality scenarios, and for nearly all classes within each modality scenario, the task of finding kelp from visual images is better without multimodal encoding. This possibly suggests that the kelp class labels are very closely tied to just the visual features rather than both visual and bathymetric information. As a result, using a feature encoding learned over both modalities is more ambiguous than using purely visual features.

4.3.3 Feature space analysis

The effect of multimodal learning can be better understood by analysing the midlayer and shared layer feature representations. Since both of these feature spaces are very high-dimensional, it can be difficult to adequately visualise them and understand the structure of the features with respect to the habitat class labels.

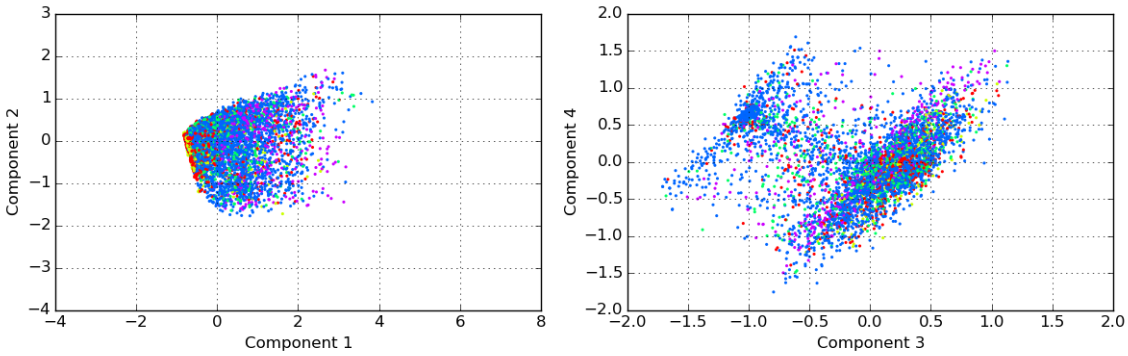
Nonetheless, PCA affords a straightforward technique to project the data into a low-dimensional space for visualisation purposes. PCA performs dimensionality reduction of the data by preserving the independent dimensions (known as principal components) which have the greatest variance. These dimensions can then be plotted to understand the primary factors of variation in the data.

Figures 4.3 and 4.4 plot the first four principal components of the midlayer features and shared layer features respectively. The first two principal components are shown on the left-hand plot, while the third and fourth components are shown on the right-hand plot. Each row in the figure corresponds to a different modality scenario, with either bathymetric features ($\mathcal{B}$), visual features ($\mathcal{V}$), or both ($\mathcal{B}+\mathcal{V}$). The points used in each plot are coloured according to the corresponding habitat label.

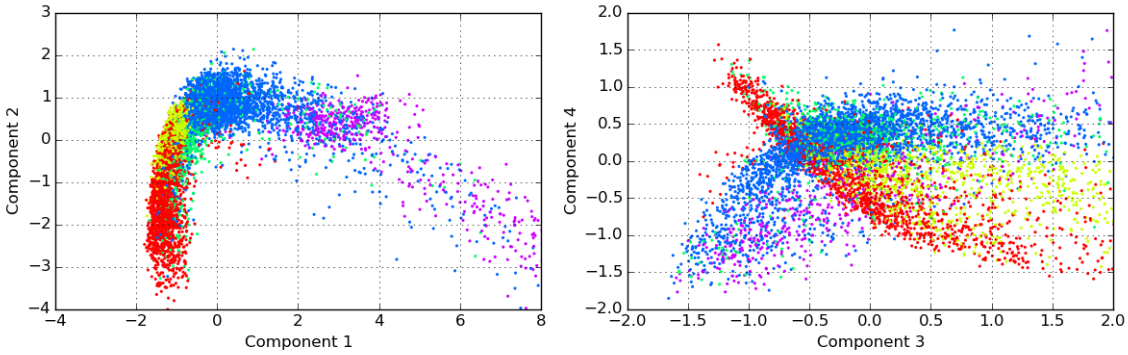
The most noticeable observation is the similarity between the visual feature space and the combined features from both modalities. Indeed, their first two principal components are nearly identical, for both the midlayer encoding and the multimodal encoding. This indicates that the predominant factors of variation (over both modalities) are in the visual data, and the addition of bathymetric features does not drastically change these principal components. This also explains why habitat classes are much easier to distinguish in visual data: because there is far more variance in the data, and as indicated in the plots, the classes are more easily separable.

The bathymetric features, in contrast, have a lower variance, for both the midlayer and shared layer features. Further, the classes are quite difficult to separate from the first four principal components in both of these cases. Given that the classification accuracy is just over 80% for bathymetric data encoded by the shared layer (Table 4.1), this would suggest that the key features are spread over a much larger number of dimensions. This is in direct contrast to the $\mathcal{V}$ and $\mathcal{B}+\mathcal{V}$ cases, in which the important structure is captured across the first few principal components.

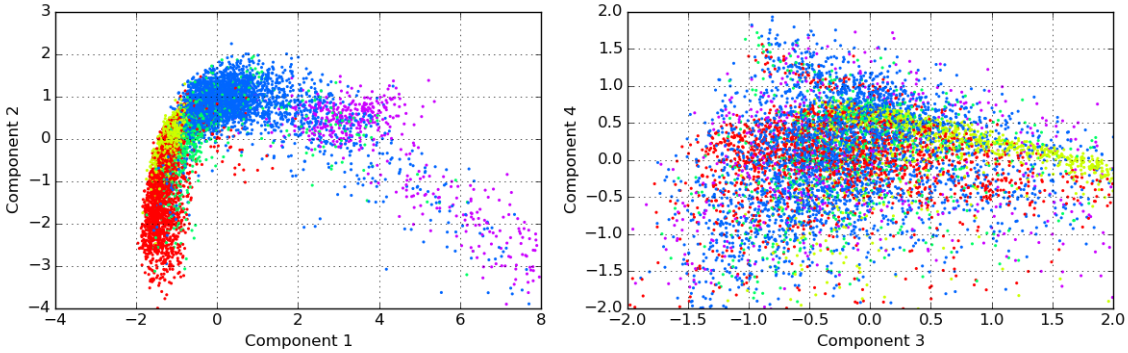
However, it is also noteworthy that for the shared layer representation, the bathymetric features are slightly more similar to the other modality cases, than for the midlayer representation. In particular, note that for the multimodal encoding, the first principal component for bathymetric features correlates more closely with the $\mathcal{V}$ and $\mathcal{B}+\mathcal{V}$ cases: it has a more similar range, and the ordering of classes (from sand to kelp) matches that of the $\mathcal{V}$ and $\mathcal{B}+\mathcal{V}$ cases. This is likely due to the fact that the shared layer projects the inputs into the same high-dimensional space: as a result, the projected bathymetry data occupies the same feature space as the projected visual data.



(a) Bathymetric features ($\mathcal{B}$) with midlayer encoding

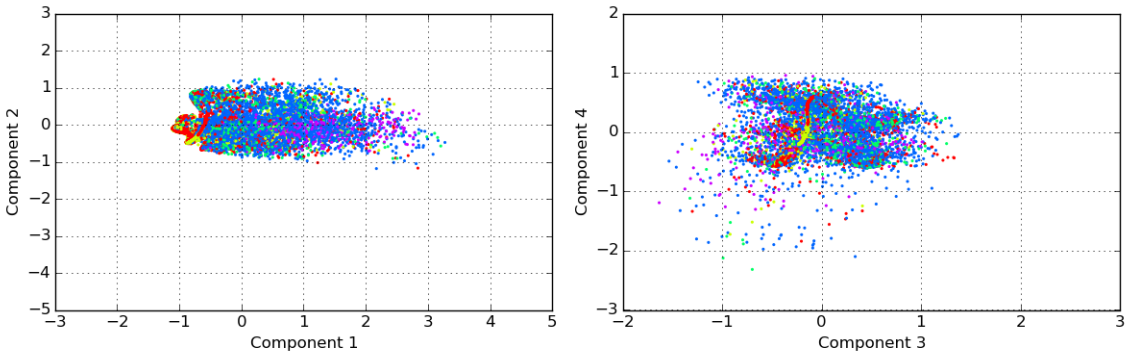


(b) Visual features ($\mathcal{V}$) with midlayer encoding

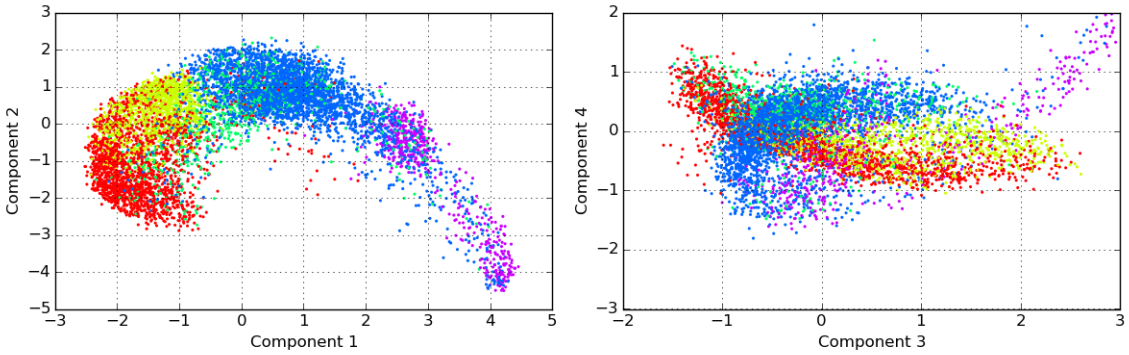


(c) Both modalities ($\mathcal{B} + \mathcal{V}$) with midlayer encoding

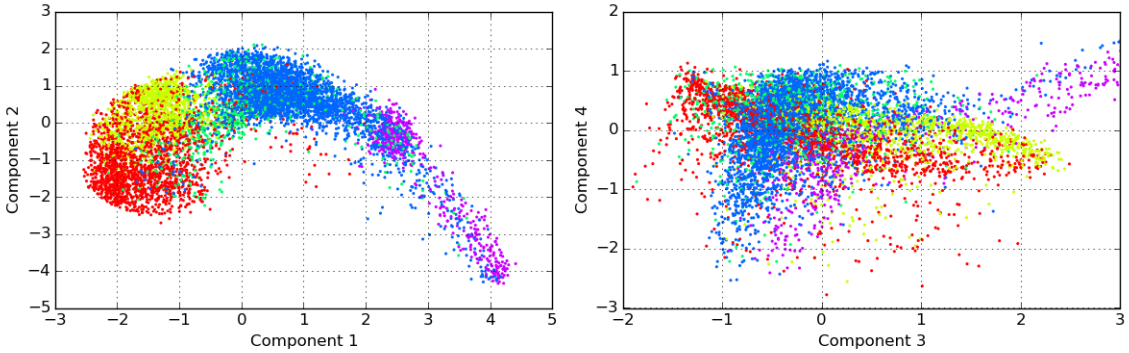
Figure 4.3 – The first four principal components for midlayer features, for all three modality scenarios. The first two components are plotted on the left, and the third and fourth components are on the right. The points are coloured by their class labels: sand (red), screw shell rubble (yellow), reef / sand interface (green), reef (blue) and kelp (purple).



(a) Bathymetric features ($\mathcal{B}$) with multimodal encoding



(b) Visual features ($\mathcal{V}$) with multimodal encoding



(c) Both modalities ($\mathcal{B} + \mathcal{V}$) with multimodal encoding

Figure 4.4 – The first four principal components for shared layer features, for all three modality scenarios. The first two components are plotted on the left, and the third and fourth components are on the right. The points are coloured by their class labels: sand (red), screw shell rubble (yellow), reef / sand interface (green), reef (blue) and kelp (purple).

4.3.4 Habitat Mapping

With the improvement in classification performance afforded by the multimodal learning model, it is possible to perform benthic habitat mapping with greater accuracy. Crucially, by performing multimodal learning prior to bathymetric classification, visual feature information is implicitly encoded into the classification process.

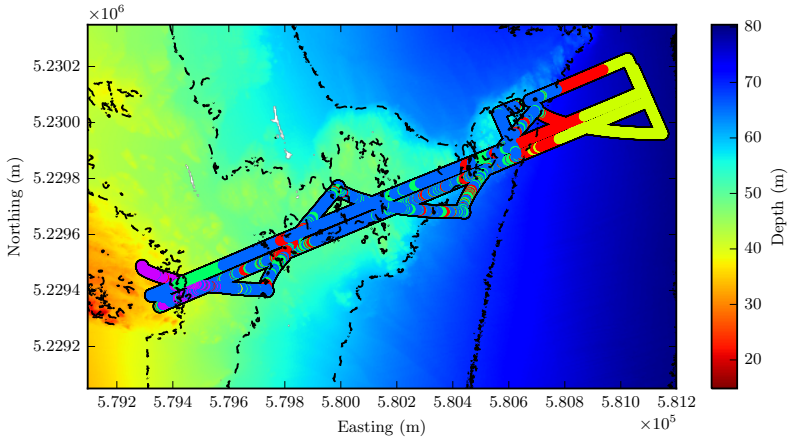
In this experiment, the multimodal approach is applied to the same habitat mapping task as in Chapter 3. As in Section 3.3.3.4, habitat mapping is performed in O'Hara Bluff. While the multimodal learning model is trained over the entire training dataset, the classifier is still trained on only the O'Hara dive data, using the multimodal encoding of the bathymetric features at each location.

Figure 4.5(a) shows the bathymetry map for O'Hara, Figure 4.5(b) contains the produced habitat map, and the corresponding class probability maps are shown in Figure 4.5(c). As with Section 3.3.3.4, the AUV trajectory is overlaid on the maps, coloured by the ground truth labels of the in-situ imagery, and the strength of the colour in the habitat map fades to white as the class probability is reduced.

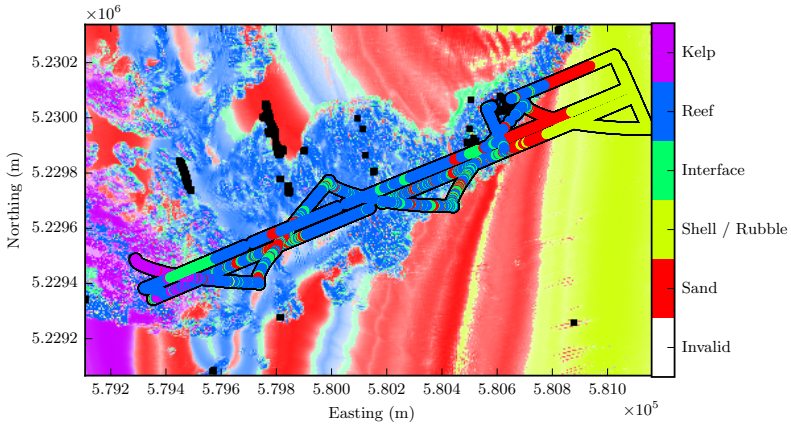
The resulting habitat map has more fine-scale variation than that obtained with the midlayer features in Chapter 3. While the depth is still important in discriminating between classes, there is a lot more variation, and the number of linear striations due to depth dependence is reduced. In particular, the flat-bottomed regions towards the north of the map are classified as sand, which is more likely than the reef label assigned by the midlayer classifier in Section 3.3.3.4.

4.3.5 Generative Sampling

In addition to classification experiments, it is important to test whether the model properly learns the distribution of the underlying data, and to determine how well it has learned the relationship between the two modalities. To test this, the model can be used to generate samples of bathymetric features, conditioned on an input image. This corresponds to the query *"What kinds of bathymetric features might be present*

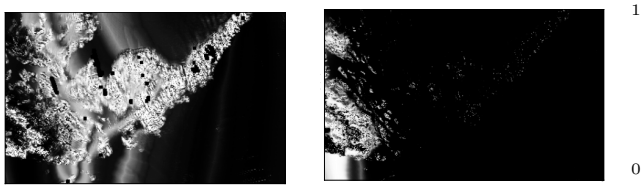


(a) Bathymetry map over the O'Hara Bluff region



(b) Habitat map using shared layer features

(i) Sand (ii) Screw Shell Rubble (iii) Reef / Sand Interface



(iv) Reef (v) Kelp

(c) Individual class probability maps using shared layer features

Figure 4.5 – Habitat mapping results for the O'Hara Bluff region using shared layer features. Each map is overlaid with the habitat labels corresponding to images taken during AUV transects in the area. The classes are sand (red), screw shell rubble (yellow), reef / sand interface (green), reef (blue), and kelp (purple). The habitat map fades to white in uncertain locations. These images are best viewed in colour.

in conjunction with this image?". If it has truly learned the relationship between the two, the resulting samples should match expert predictions on the co-occurrence of bathymetric features with the habitat class observed in the conditioning image (for example, that kelp is likely to be found in shallow, rugose terrain, or that screw shell rubble usually coincides with deeper, flatter areas).

This experiment is performed using the procedure described in Section 4.2. By continuously running the iterative procedure, with 10 iterations between each generated sample, we can obtain samples similar to those shown in Figure 4.6. Here, the model uses a set of input images, each from a different habitat class (top row) to generate samples of encoded depth features and bathmetric patches, respectively (following rows). As the depth features are encoded as a 1-of-k with Gaussian falloff, each depth "signal" should be interpreted as an activation function, where a high activation value suggests a higher likelihood of observing that depth.

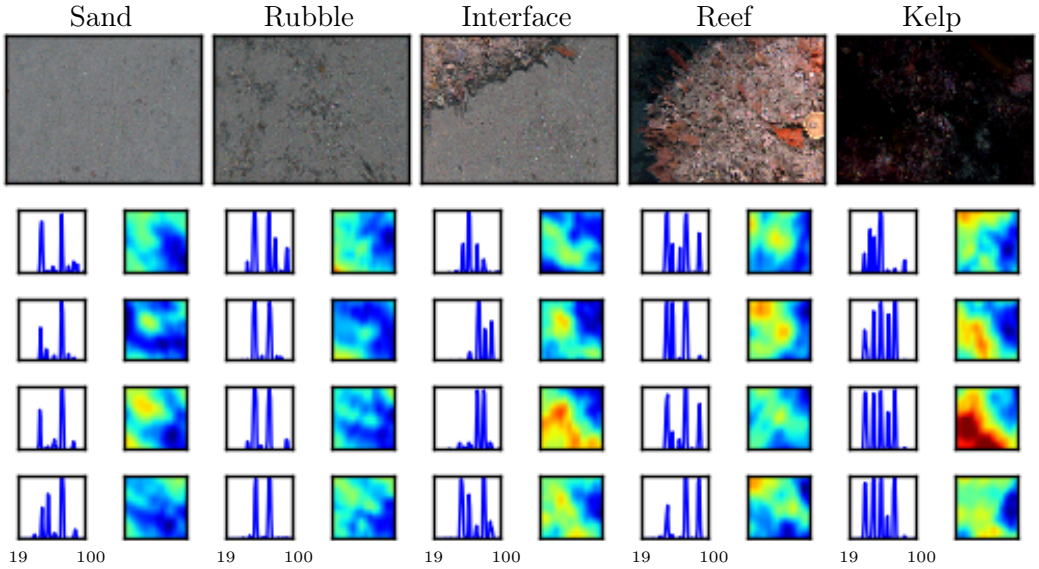


Figure 4.6 – Bathymetry samples obtained from the learned data-generating distribution, conditioned on the input image. For each input image representing a single habitat class (top row), the subsequent rows display different examples of model-generated samples of $\mathcal{B}_0$ in encoded form (left) and $\mathcal{B}_l$ patches (right). Shallower regions are represented as red in the patches, and the $\mathcal{B}_0$ signal should be interpreted as an 'activation function' over the depth range $19 - 100$m).

To quantitatively analyse the results, the model was used to generate 1000 such

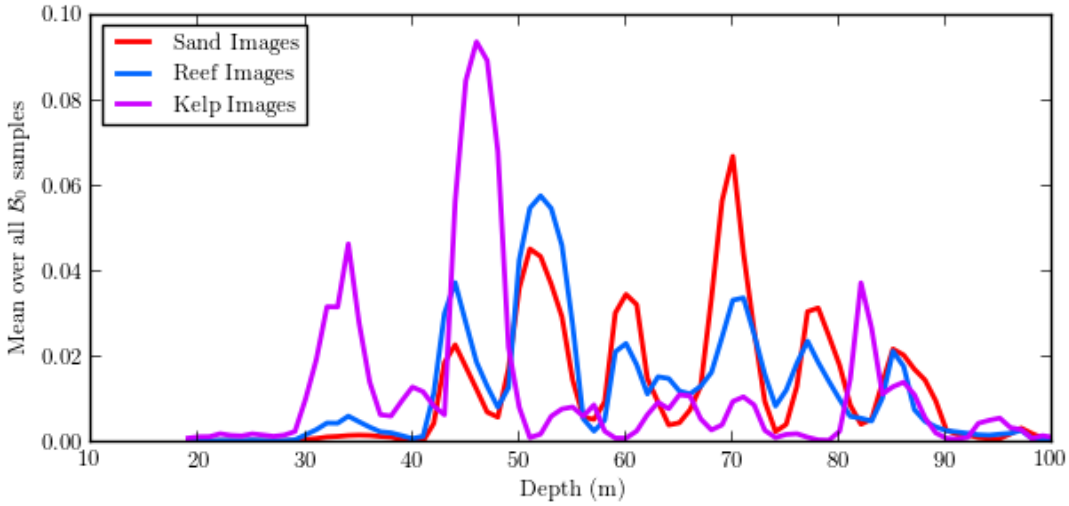


Figure 4.7 – Average of encoded $\mathcal{B}_0$ (depth) samples conditioned on every image in O'Hara Bluff (1000 samples per image). Samples are grouped by the class of the image used to generate them, and a few key classes are shown here.

samples of $\mathcal{B}_0$ and $\mathcal{B}_l$ for every image in the O'Hara Bluff region. The images were then grouped by class label, and the depth samples averaged over each class to show the distribution over the entire depth range. The mean depth samples for sand, reef, and kelp classes are shown in Figure 4.7. Additionally, the rugosity was computed for each generated bathymetry patch, and the mean and standard deviation over each class are shown in Table 4.2.

The results suggest that the model is learning the underlying data distribution. The sampled bathymetric patches are, on average, smoother for the sand classes and more rugose for the reef and kelp images (Table 4.2). Similarly, while the kelp image activates depth features at the shallower end of the range, deeper areas are activated for sand and reef (Fig. 4.7). It is also important to note that the variation within each class is indicative of the spatial distribution of the class. For example, while the mean depth sample for kelp images only has a few large peaks, mostly in shallow areas, the corresponding signal for sand or reef is spread over a larger depth range.

Table 4.2 – Rugosity of $\mathcal{B}_l$ (bathymetry patch) samples, conditioned on every image in O'Hara Bluff (1000 samples per image), grouped by class.

	Sand	Shell rubble	Interface	Reef	Kelp
Mean	1.0773	1.0804	1.0989	1.1218	1.1846
Std	0.0184	0.0307	0.0365	0.0478	0.0580

4.4 Summary

This chapter proposed a multimodal learning model to capture the relationship between the visual image data and co-located bathymetric features. Using a similar architecture to previous work [69, 85], the proposed approach involves training a shared layer on the concatenation of the high-level features from each modality. This represents a novel application of multimodal learning algorithms to visual and remotely sensed marine data.

Classification was performed with the proposed model for various modality combinations, emulating the situations in which either visual or bathymetric data is unavailable. Results with the proposed model demonstrate that by performing multimodal learning beforehand, classification accuracy is improved, regardless of which modalities are available at classification time. In particular, for the habitat mapping scenario, in which visual data is unavailable, classification of features extracted by the multimodal learning approach was found to be 11% more accurate than with the bathymetric features directly.

Generative sampling was also performed, by using the model to obtain bathymetric samples conditioned on an input image. The sampled bathymetric samples were in line with expert predictions: the rugosity of the bathymetric patches generated from reef and kelp images was much greater than from images of sand and screw shell rubble, and the generated depth signals were at lower depths for kelp than for reef and sand classes. The results demonstrated that the model can properly capture the underlying data distribution, and understand the relationship between the two modalities.

Chapter 5

Extending multimodal learning with gated models

In this chapter, we build upon the multimodal architecture of the previous chapter with more sophisticated gated models, based on a mixture of RBMs (MixRBM). This model can capture the one-to-many relationship between the visual and bathymetric modalities, and a number of novel improvements are presented to facilitate more sophisticated learning and inference tasks, such as image-based queries. Section 5.1 discusses the motivation behind the proposed model, and Section 5.2 introduces the gated model used in this chapter. Section 5.3 describes the learning algorithms for the model, and proposes heuristics to avoid having to specify the number of mixture components. Section 5.4 then outlines the inference procedures under the model, including novel derivations that enable the clustering of single modalities, and algorithms to predict visual features from bathymetric data and handle image-based queries. Finally, Section 5.5 outlines experiments with a toy dataset and marine data, validating the above inference procedures.

5.1 Motivation

One important consideration for this application is that bathymetry is a coarser sensor modality: a single 'type' of feature may correspond to many 'types' of visual features. More specifically, the conditional distribution of visual features given bathymetric features may be highly multimodal.

Unfortunately, the multimodal model presented in the previous chapter does not explicitly consider this factor. While the shared layer can sample bathymetric features given visual data (Section 4.3.5), the more interesting task is the inverse problem, predicting visual features in unseen areas from the known bathymetry. This would enable planning queries based on image features, indicating where an input image, or a particular set of features, are likely to be observed. This capability has not previously been introduced in either the robotics or habitat mapping communities.

The standard multimodal model could achieve this by generating a very large number of visual samples conditioned on the bathymetry, and using these to approximate the conditional distribution of visual features given the bathymetry. However, as will be discussed in this section, this is difficult because the conditional distribution has several modes. Instead, it is desirable for the model to provide a principled way in which to select a conditional mode: the model could provide a summary of the different types of visual features that are observed, and associated probabilities of observing them.

This chapter proposes using a gated mixture of RBMs model [67], which is better equipped to handle the 'one-to-many' relationship between the two modalities. In the gated model, the joint distribution over both modalities is conditioned on a latent indicator variable. This effectively learns multiple RBM components under the same framework, with the indicator variable switching between them on the fly.

The architecture employed by this model is similar to that presented in the previous chapter. As in Figure 5.1, the visual and bathymetric features are concatenated at the mid layer, which are then passed into the gated MixRBM model (the shared layer).

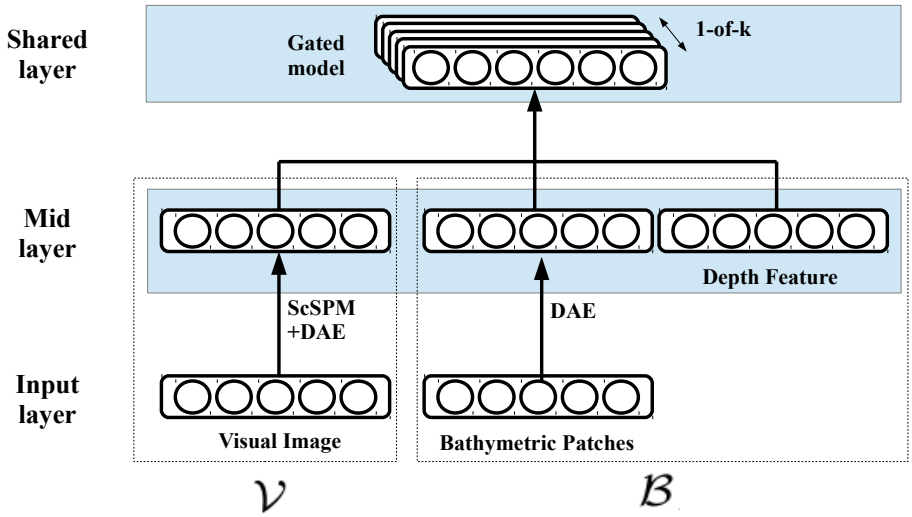


Figure 5.1 – Schematic showing the gated model architecture. Features from both modalities are concatenated in the mid layer, and then passed into the shared layer. The one-of-k indicator variable indexes a single RBM component from the Mixture of RBMs model.

Thus, the mid layer features are identical to Chapter 4, and the only change is the additional complexity in the shared layer.

5.2 Gated Boltzmann Machines and mixtures of RBMs

A Gated Boltzmann Machine (GBM) [63, 67] is a probabilistic graphical model comprised of a set of binary visible variables $\mathbf{x} \in \{0,1\}^{n_x}$, hidden variables $\mathbf{h} \in \{0,1\}^{n_h}$, and gating, or *conditioning*, variables $\mathbf{z} \in \{0,1\}^{n_z}$. With the addition of the gating variables, the graphical model of the GBM is a tripartite graph, as compared to the bipartite graph of the standard RBM (described in Section 2.3). The visible variables $\mathbf{x}$ represent the input data, which in this case is the concatenation of the visual features $\mathbf{x}_{\mathcal{V}}$ and bathymetric features $\mathbf{x}_{\mathcal{B}}$.

The model captures the joint relationship between the visible and hidden units, con-

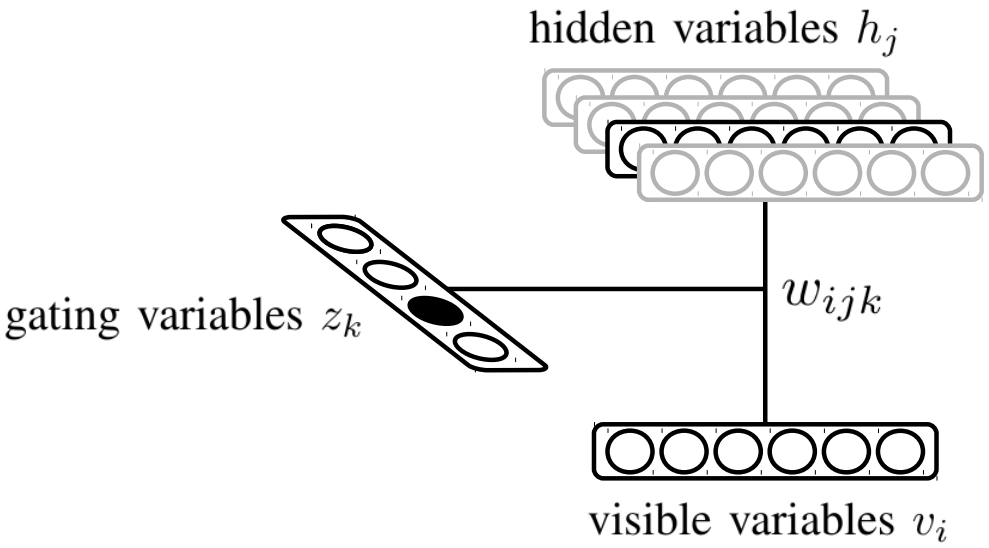


Figure 5.2 – Graphical representation of a gated mixture of RBMs. By turning on the k^{th} gating variable, the model uses the component RBM whose parameters are contained in the k^{th} slice of the parameter tensor, thereby using the corresponding set of hidden units.

ditioned on the gating variables, using an energy function:

$$\begin{aligned} E(\mathbf{x}, \mathbf{h} \mid \mathbf{z}) &= -\sum_{ijk} w_{ijk} x_{ik} h_{jk} z_k \\ &\quad -\sum_{ik} a_{ik} x_{ik} - \sum_{jk} b_{jk} h_{jk} \\ p(\mathbf{x}, \mathbf{h} \mid \mathbf{z}) &= \frac{e^{-E(\mathbf{x},\mathbf{h}|\mathbf{z})}}{Z} \end{aligned} \tag{5.1}$$

where $\mathbf{W} = [w_{ijk}]$ is the weights tensor, $\mathbf{a} = [a_{ik}]$ and $\mathbf{b} = [b_{jk}]$ are the visible and hidden biases respectively, and $Z = \sum_{\mathbf{x},\mathbf{h}} e^{-E(\mathbf{x},\mathbf{h}|\mathbf{z})}$ is the partition function.

If the gating variables are constrained to be a 'one-of-k' (i.e. $\mathbf{z} \in \{0,1\}^{n_z}, \sum_k z_k = 1$), then each possible value for $\mathbf{z}$ indexes a single 2D slice of $\mathbf{W}$ and a 1D slice of each bias matrix. This forms a mixture of Restricted Boltzmann Machines (RBMs) [67], where $\mathbf{z}$ is a mixture indicator variable used to select one of n_z RBM components (Figure 5.2), each with separate weights and biases.

The model yields the following conditional expressions:

$$\begin{aligned} p\left(h_j = 1 \mid \mathbf{x}, \mathbf{z}_k = 1\right) &= \operatorname{sigm}\left(b_{jk} + \sum_i w_{ijk} x_i\right) \\ p\left(x_i = 1 \mid \mathbf{h}, \mathbf{z}_k = 1\right) &= \operatorname{sigm}\left(a_{ik} + \sum_j w_{ijk} h_j\right) \end{aligned} \tag{5.2}$$

where $\operatorname{sigm}(x) = (1 + e^{-x})^{-1}$ is the element-wise logistic sigmoid function. For a specific set value of the gating variable $\mathbf{z}$, these equations are equivalent to a standard RBM using the k^{th} slice of each parameter tensor.

The probability of an input vector $\mathbf{x}$ can be obtained by marginalising the joint density $p(\mathbf{x}, \mathbf{h})$ over the hidden units:

$$\begin{aligned} F\left(\mathbf{x} \mid \mathbf{z}_k = 1\right) &= -\sum_i a_{ik} x_{ik} - \sum_j \log\left(1 + e^{b_{jk} + \sum_i w_{ijk} x_{ik}}\right) \\ p\left(\mathbf{x} \mid \mathbf{z}_k = 1\right) &= \frac{\sum_{\mathbf{h}} e^{-E(\mathbf{x}, \mathbf{h} \mid \mathbf{z}_k = 1)}}{Z} = \frac{e^{-F(\mathbf{x} \mid \mathbf{z}_k = 1)}}{Z} \end{aligned} \tag{5.3}$$

where the expression $F\left(\mathbf{x} \mid \mathbf{z}_k = 1\right)$ is known as the *free energy* of a visible vector under the k^{th} RBM component. Unfortunately, the partition function Z is intractable, which means that the RBM can only compute *unnormalised* probabilities.

However, for a given input vector, the mixture probabilities can be determined exactly according to the free energy:

$$p\left(z_k = 1 \mid \mathbf{x}\right) = \frac{e^{-F(\mathbf{x} \mid \mathbf{z}_k = 1)}}{\sum_k e^{-F(\mathbf{x} \mid \mathbf{z}_k = 1)}} \tag{5.4}$$

Note that the denominator in (5.4) is tractable, and is linear in the number of mixture components n_z.

5.3 Learning

Given a set of training vectors $\{\mathbf{x}^{(1)}, \cdots, \mathbf{x}^{(N)}\}$, MixRBM models are usually trained to maximise the mean log probability of the data, $L = \frac{1}{N}\sum_{n=1}^{N} \log p(\mathbf{x}^{(n)})$, using SGD, in a similar fashion to RBMs. The gradient of L with respect to the parameters Θ is given by:

$$\frac{\partial L}{\partial \Theta} = N\mathbb{E}\left[\frac{\partial E(\mathbf{x}, \mathbf{h}, \mathbf{z})}{\partial \Theta}\right] - \sum_{n=1}^{N} \mathbb{E}\left[\frac{\partial E(\mathbf{x}^{(n)}, \mathbf{h}, \mathbf{z})}{\partial \Theta}\Big|\mathbf{x}^{(n)}\right] \tag{5.5}$$

As with RBMs, the second expectation can be estimated using Gibbs sampling, but the first term is intractable. As a result, the Maximum Likelihood gradients are again approximated using the Contrastive Divergence (CD) algorithm [67]. The procedure is a slight modification to CD for RBMs, and is shown in Algorithm 5.1. The main difference is that for each input vector, a specific mixture component is selected by sampling over the mixture probabilities, and the selected component is used for a single iteration of Gibbs sampling. This process is repeated for each iteration, which means that the selected component may be different for the positive and negative statistics of CD learning.

Here, $\mathbf{W}_k$, $\mathbf{a}_k$, and $\mathbf{b}_k$ are slices of the parameter tensors corresponding to the k^{th} component RBM.

5.3.1 Cluster Heuristics

In the original formulation of the Mixture of RBMs model [67], an additional temperature parameter T was used to scale the free energies before computing the mixture probabilities in Equation 5.4. This was a necessary inclusion due to the fact that the free energy is an *unnormalised* quantity, and helped to prevent the scenario of a single mixture component having a high responsibility for most of the dataset.

We instead solve this by introducing heuristics to add and remove components during training. This also has the added benefit that the number of mixture components does not need to be specified beforehand. The heuristics are based on the simple

Algorithm 5.1: Contrastive Divergence (CD-1) training for a gated mixture of Restricted Boltzmann Machines

$\frac{\partial L}{\partial \mathbf{W}} \Leftarrow \mathbf{0}$, $\frac{\partial L}{\partial \mathbf{a}} \Leftarrow \mathbf{0}$, $\frac{\partial L}{\partial \mathbf{b}} \Leftarrow \mathbf{0}$
for $i = 0$ to N **do**
  $\mathbf{x}_+ \Leftarrow$ training sample i
  Sample $\mathbf{z} \sim p(\mathbf{z}_k = 1|\mathbf{x}_+)$. Let the selected component be indexed by ℓ_+.
  Sample $\mathbf{h}_+ \sim p_{\ell_+}(\mathbf{h}|\mathbf{x}_+)$
  Sample $\mathbf{x}_- \sim p_{\ell_+}(\mathbf{x}|\mathbf{h}_+)$
  Sample $\mathbf{z} \sim p(\mathbf{z}_k = 1|\mathbf{x}_-)$. Let the selected component be indexed by ℓ_-.
  Sample $\mathbf{h}_- \sim p_{\ell_-}(\mathbf{h}|\mathbf{x}_-)$
  $\frac{\partial L}{\partial \mathbf{W}_{\ell_+}} \Leftarrow \frac{\partial L}{\partial \mathbf{W}_{\ell_+}} + \mathbf{x}_+\mathbf{h}_+^T/N$
  $\frac{\partial L}{\partial \mathbf{a}_{\ell_+}} \Leftarrow \frac{\partial L}{\partial \mathbf{a}_{\ell_+}} + \mathbf{x}_+/N$
  $\frac{\partial L}{\partial \mathbf{b}_{\ell_+}} \Leftarrow \frac{\partial L}{\partial \mathbf{b}_{\ell_+}} + \mathbf{h}_+/N$
  $\frac{\partial L}{\partial \mathbf{W}_{\ell_-}} \Leftarrow \frac{\partial L}{\partial \mathbf{W}_{\ell_-}} - \mathbf{x}_-\mathbf{h}_-^T/N$
  $\frac{\partial L}{\partial \mathbf{a}_{\ell_-}} \Leftarrow \frac{\partial L}{\partial \mathbf{a}_{\ell_-}} - \mathbf{x}_-/N$
  $\frac{\partial L}{\partial \mathbf{b}_{\ell_-}} \Leftarrow \frac{\partial L}{\partial \mathbf{b}_{\ell_-}} - \mathbf{h}_-/N$
end for

intuition that it is undesirable for a mixture component to be responsible for a very large or very small fraction of the dataset.

5.3.1.1 Removing clusters

Based on experiments with the model, it is clear that even if the specified number of clusters is much larger than the expected number, the model naturally uses fewer components to describe the data. An effective approach is to monitor the mixture responsibility $p\left(z_k = 1 \mid \mathbf{x}\right)$ of a cluster k, and remove the cluster if the mean mixture responsibility (over the entire dataset) drops below a threshold.

5.3.1.2 Splitting clusters

In a similar fashion, a cluster can be split if its mean mixture responsibility exceeds a certain threshold. This helps to prevent the scenario where a single cluster is used to describe a large proportion of the dataset. When splitting a cluster, the new cluster

parameters are copied directly from the existing cluster. Experiments show that after a few parameter updates, the two identical clusters diverge to capture different parts of the input dataset.

5.4 Inference

This section describes the different inference tasks that can be accomplished using the model. This includes novel derivations that enable single-modality inference, including the ability to predict visual features from bathymetry data, leading to the ability to handle image-based queries.

5.4.1 Joint Sampling

In a standard RBM, the visible and hidden units form a Markov chain, and Markov Chain Monte Carlo (MCMC) techniques can generate samples of the input distribution. From a random visible configuration, multiple iterations of block Gibbs sampling are applied: the hidden and visible variables are sequentially sampled from each other in an alternating fashion based on their conditional distributions.

In a Mixture of RBMs, we first select a mixture component using $p\left(z_k = 1 \mid \mathbf{x}\right)$, and then perform an iteration of Gibbs sampling using the corresponding RBM component.

5.4.2 Conditional Sampling and Prediction

The benefit of the gated model is that it can be used to predict the visual features $\mathbf{x}_{\mathcal{V}}$ in unobserved areas, conditioned on the bathymetric features $\mathbf{x}_{\mathcal{B}}$. Without loss of generality, we assume that the inference task is to predict / sample visual features given bathymetric features, though the reverse can also be achieved through a similar procedure.

Algorithm 5.2: Predicting visual features from bathymetry

for $k = 1$ to n_z **do**
  Initialise the mid layer feature vector with zeros for the visual features, $\mathbf{x} = [\mathbf{x}_\mathcal{B}; \mathbf{x}^*_\mathcal{V}] = [\mathbf{x}_\mathcal{B}; 0; 0; ...; 0]$.
  while not converged **do**
    Compute $\mathbb{E}_k[\mathbf{h}|\mathbf{x}] = p(\mathbf{h}|\, \mathbf{x}, \mathbf{z}_k = 1)$.
    Compute $\mathbb{E}_k[\mathbf{x}_\mathcal{V}|\mathbf{h}] = p(\mathbf{x}_\mathcal{V}|\, \mathbf{h}{=}\mathbb{E}_k[\mathbf{h}|\mathbf{x}], \mathbf{z}_k = 1)$
    if $\|\mathbf{x}^*_\mathcal{V} - \mathbb{E}_k[\mathbf{x}_\mathcal{V}|\mathbf{h}]\| < \epsilon$ **then**
      converged
    else
      $\mathbf{x}^*_\mathcal{V} \Leftarrow \mathbb{E}_k[\mathbf{x}_\mathcal{V}|\mathbf{h}]$, $\mathbf{x} \Leftarrow [\mathbf{x}_\mathcal{B}; \mathbf{x}^*_\mathcal{V}]$
    end if
  end while
  $\mathbb{E}_k[\mathbf{x}_\mathcal{V}|\mathbf{x}_\mathcal{B}] \Leftarrow \mathbf{x}^*_\mathcal{V}$.
end for

This is achieved using a mean field approximation (Algorithm 5.2). For each mixture component, this approximation involves using the input values to compute the mean hidden activations $\mathbb{E}[h_j|\mathbf{x}, z_k = 1] = p\,(h_j = 1 \mid \mathbf{x}, \mathbf{z}_k = 1)$, which are then in turn used to compute the conditional expectations $\mathbb{E}[\mathbf{x}_\mathcal{V}|\mathbf{h}, z_k = 1]$. This process can be iterated until convergence, yielding the conditional expectation of $\mathbf{x}_\mathcal{V}$ given $\mathbf{x}_\mathcal{B}$ under the k^{th} component, which we denote as $\mathbb{E}_k[\mathbf{x}_\mathcal{V}|\mathbf{x}_\mathcal{B}]$. Experiments show that a single iteration is enough to yield a good conditional estimate.

The bathymetry-only mixture responsibilities can then be approximated as follows.

$$p\,(z_k = 1 \mid \mathbf{x}_\mathcal{B}) \approx \frac{e^{-F(\mathbf{x}_\mathcal{B}, \mathbf{x}_\mathcal{V} = \mathbb{E}_k[\mathbf{x}_\mathcal{V}|\mathbf{x}_\mathcal{B}], \mathbf{z}_k = 1)}}{\sum_k e^{-F(\mathbf{x}_\mathcal{B}, \mathbf{x}_\mathcal{V} = \mathbb{E}_k[\mathbf{x}_\mathcal{V}|\mathbf{x}_\mathcal{B}], \mathbf{z}_k = 1)}} \tag{5.6}$$

That is, we use each component RBM to fill in the missing visual feature dimensions with their conditional expectations, compute the free energies given these 'best-case' scenarios, and then normalise the probabilities over all mixture components. Effectively, this is equivalent to appxoimating a highly multimodal distribution by the set of means of all of the modes, which means that the variance of $p\,(\mathbf{x}_\mathcal{V} \mid \mathbf{x}_\mathcal{B})$ within each mixture component is neglected. Thus, for a given bathymetric feature vector,

the model is able to predict k different options for the visual features, along with associated probabilities of each of them occurring.

If we want to generate samples from the conditional distribution $p(\mathbf{x}_\mathcal{V} \mid \mathbf{x}_\mathcal{B})$ rather than computing the expectation, we can sample a component RBM with probability $p(z_k = 1 \mid \mathbf{x}_\mathcal{B})$ approximated by the above procedure, then perform Gibbs sampling with the selected Markov chain.

5.4.3 Image-based queries

A key contribution of this chapter is the ability to handle image-based queries. Given a region of interest, visual features can be predicted from the bathymetry and compared to a query image to produce a utility map over the whole region. This can then be used by a planning algorithm to explore areas where similar images are likely to be observed.

The query-by-image procedure is as follows. First, for each point in the region of interest, we predict the visual features from the local bathymetry (i.e. compute the conditional expectation $\mathbb{E}_k[\mathbf{x}_\mathcal{V} \mid \mathbf{x}_\mathcal{B}]$ according to each mixture component k), and compute the marginal mixture responsibilities $p(\mathbf{z} \mid \mathbf{x}_\mathcal{B})$. We then define a utility function $\mathcal{U}$, which acts as a proxy for the likelihood of observing the query image given the bathymetry at a particular location. The utility at a particular location is based on the similarity between the query image to each of the n_z predicted images, scaled by the associated mixture probabilities:

$$\mathcal{U} = \sum_{k=1}^{n_z} p(z_k = 1 \mid \mathbf{x}_\mathcal{B}) \; \mathcal{S}(\mathbf{x}_{\mathcal{V}q}, \mathbb{E}_k[\mathbf{x}_\mathcal{V} \mid \mathbf{x}_\mathcal{B}]) \tag{5.7}$$

where $\mathbf{x}_{\mathcal{V}q}$ is the midlayer visual feature vector for the query image, and $\mathcal{S}(\mathbf{u}, \mathbf{v})$ is a metric computing the similarity between $\mathbf{u}$ and $\mathbf{v}$. In this work, we use the normalised cross-correlation metric, given by $\mathcal{S}(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\| \|\mathbf{v}\|}$.

5.4.4 Classification

When using a standard RBM to model input data, the hidden activations can be used in a linear classifier. With a Mixture of RBMs, there are a number of options for features that can be extracted for classification [67]:

- **Mixture responsibilities**: these are often a good low-dimensional feature set, since the model naturally uses different mixture components for different parts of the input space.
- **Most probable mixture component**: the single component oftens provides some information about the class label.
- **Hidden activations (all)**: for a given data vector, we obtain hidden unit activations for all mixture components and stack them into a single vector.
- **Hidden activations (single mixture)**: alternatively, we obtain the hidden activations for the most probable mixture, and set the activations of all other mixtures to zero.
- **Hidden activations (scaled)**: we obtain hidden unit activations for all mixture components, and then scale each unit by its corresponding mixture probability.

When modalities are missing, classification features can be extracted as follows. Firstly, the missing modalities can be "reconstructed" using their conditional expectations according to each mixture component, and each of these reconstructions can be encoded using the same mixture component to compute the corresponding hidden unit activations. These can be stacked together to form the "hidden activations (all)" features. The single-modality mixture responsibilities can be computed from the conditional expectations according to the procedure described in Section 5.4.2. The remaining feature options are obtained by either selecting hidden activations from the most likely component, or scaling the activations by the corresponding mixture probabilities.

5.5 Experiments

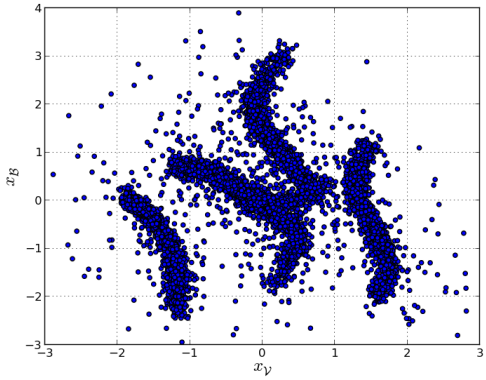
This section details the experiments with the gated multimodal model, performed on both simulated data and the multimodal marine dataset used in the previous chapters. For the marine dataset, the preprocessing steps and midlayer architecture are identical to the standard multimodal model described in Chapter 4. For the shared layer, the total number of hidden units is kept the same as the standard model (2000), with a maximum of 20 mixture components, each containing 100 hidden units. Experiments were performed with 10, 20, 50 and 100 mixture components, and the selected value exhibited the best clustering and classification performance.

For the first 10 epochs of learning, we only enable a single mixture component: this ensures that the model starts with a reasonable representation of the data before attempting to split the data into clusters. Following this stage, the cluster heuristics ensure that most, if not all, of the 20 available cluster components are utilised. To encourage robustness to missing modalities, we train the model in a denoising fashion: for each training vector, we either mask one of the modalities or utilise the full input vector, each with equal probability. This has a similar effect to the denoising training criterion of the standard model.

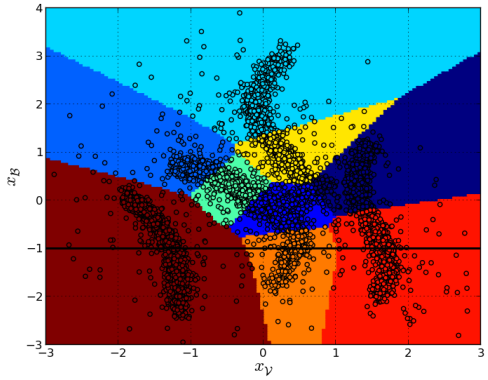
The gated model was written in Python using the pylearn2 library, and took approximately 2.5 days to train on a NVIDIA GTX 590 GPU.

5.5.1 Toy Experiments

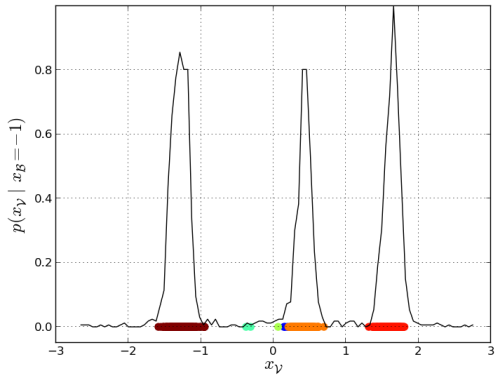
We introduce a two-dimensional toy dataset (Figure 5.3(a)) to illustrate the operation the two models and highlight their differences. While it is highly simplified compared to our real multimodal dataset, it is designed to share one key characteristic: the fact that the conditional distribution of visual features (represented by dimension $x_{\mathcal{V}}$) given bathymetric features (dimension $x_{\mathcal{B}}$) can be highly multimodal. The data was created by generating polynomial curve segments from random coefficient values with additive Gaussian noise. We train a standard RBM and the gated Mixture of RBMs



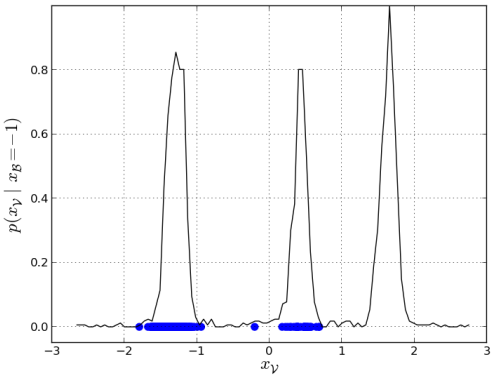
(a) The 2D toy dataset generated for this problem. The dataset is designed such that the conditional distribution $p(x_{\mathcal{V}}|x_{\mathcal{B}})$ can be highly multimodal.



(b) Input data coloured according to most probable mixture component



(c) Conditional samples generated by Mixture of RBMs (coloured by corresponding mixture component), overlaid on the conditional distribution



(d) Conditional samples generated by standard RBM, overlaid on the conditional distribution

Figure 5.3 – Clustering and sampling results for the gated model on a toy dataset

on the toy data, and perform various experiments to demonstrate the differences between the standard and gated options (Figure 5.3).

To visualise the distributions learned by each model, we first generate samples from the model, initialising the input data to random training points and performing Gibbs sampling repeatedly. From Figures 5.3(d) and 5.3(c), we observe that both models learn very similar distributions.

The key differences between the two are showcased in Figures 5.3(b) to 5.3(d). Firstly, as demonstrated by Figure 5.3(b), the gated model utilises different component RBMs to model different parts of the dataset, which means that the data can be clustered in an unsupervised fashion. Secondly, the gated model is better equipped to perform conditional sampling given a highly multimodal conditional distribution. This is shown in Figures 5.3(c) and 5.3(d), which demonstrate the result of sampling from the conditional distribution $p(x_{\mathcal{V}} \,|\, x_{\mathcal{B}}{=}{-1})$ (the line marked in Figure 5.3(b)). Since neither model can analytically determine this conditional distribution, it is approximated using the histogram of all points within $\delta = 0.05$ of the setpoint value $x_{\mathcal{B}} = -1$.

In this scenario, despite the highly multimodal conditional distribution, the gated model can produce reasonable samples, and represents each mode with a different component RBM (Figure 5.3(c)). The samples are primarily obtained using the magenta, orange, and red mixture components, as these have the greatest probability at the given setpoint, but some of the other components are also represented with nonzero probability. In contrast, Figure 5.3(d) shows the same result with a standard RBM, by initialising the missing $x_{\mathcal{V}}$ value to zero and performing a number of iterations of Gibbs sampling. With this approach, the Gibbs chain is not always able to mix between modes of the conditional distribution.

In practice, this drawback could be addressed by initialising the missing dimension randomly and repeating the process a number of times, but this scales exponentially with the number of missing dimensions. In contrast, the corresponding inference procedure for the gated model is linear in the number of mixture components.

These results illustrate the key benefits of a gated model as compared to a standard model. In addition to unsupervised clustering of the input data, the model can be used to tractably generate conditional samples from explicit regions of our highly multimodal distribution. In contrast with a standard model, the gated model can map a bathymetric feature to multiple options simultaneously rather than a single mode / label. In addition to sampling effectively from a highly multimodal conditional distribution, the model is able to select a mode in a principled way.

5.5.2 Classification

In this section, classification is performed on the marine dataset, by using the gated model to generate the features described in Section 5.4.4. The performance with the different feature scenarios is compared with the multimodal model from Chapter 4 as well as with the midlayer features (Chapter 3).

The classification accuracies are shown in Table 5.1. From the results with $\mathbf{z}$, we can observe that the most probable cluster component itself holds a lot of information about the habitat label, with 77% accuracy with both modalities. However, this is much lower for bathymetric data, indicating the difference in structure between the two modalities. Converting the one-hot vector to a vector of mixture probabilities yields a small improvement in performance for all scenario combinations. Using all hidden features, the classification performance of the gated and non-gated models are quite similar, and represent an improvement over the baseline for all combinations of modalities. Using the hidden components from just a single component, or scaling the hiddens by the mixture probabilities, means a much poorer result for the bathymetry scenario, and a slightly poorer result for the other scenarios. This supports the hypothesis proposed in Chapter 4 through PCA analysis: that for the multimodal encoding of bathymetry data, the entire set of hidden features are necessary to achieve good classification performance.

As with the standard multimodal model, the gated model performs very well in the habitat mapping scenario (with only bathymetric data available), yielding an improvement of over 10% compared to the baseline.

5.5.3 Precision and recall analysis

As with Chapter 4, this section analyses the precision-recall curves of each class for each modality scenario, in order to paint a complete picture of the strengths and weaknesses of the classifiers used.

The precision-recall curves are shown in Figure 5.4. Each row refers to a separate

Table 5.1 – Classification accuracy (%) for various input modality combinations

		Modalities		
Model	Features	$\mathcal{B}$ and $\mathcal{V}$	$\mathcal{B}$ only	$\mathcal{V}$ only
Baseline	Midlayer	83.05	72.57	79.98
DAE + LR	$p(\mathbf{h}\|\mathbf{x})$	87.43	81.23	80.71
MixRBM + LR	$p(\mathbf{h}\|\mathbf{x})$ (all)	87.88	82.66	81.81
	$p(\mathbf{h}\|\mathbf{x})$ (single)	85.87	73.64	79.75
	$p(\mathbf{h}\|\mathbf{x})$ (scaled)	86.41	76.21	80.76
	$p(\mathbf{z}\|\mathbf{x})$	78.42	64.14	73.13
	$\mathbf{z}$ (one hot)	77.83	61.61	71.51

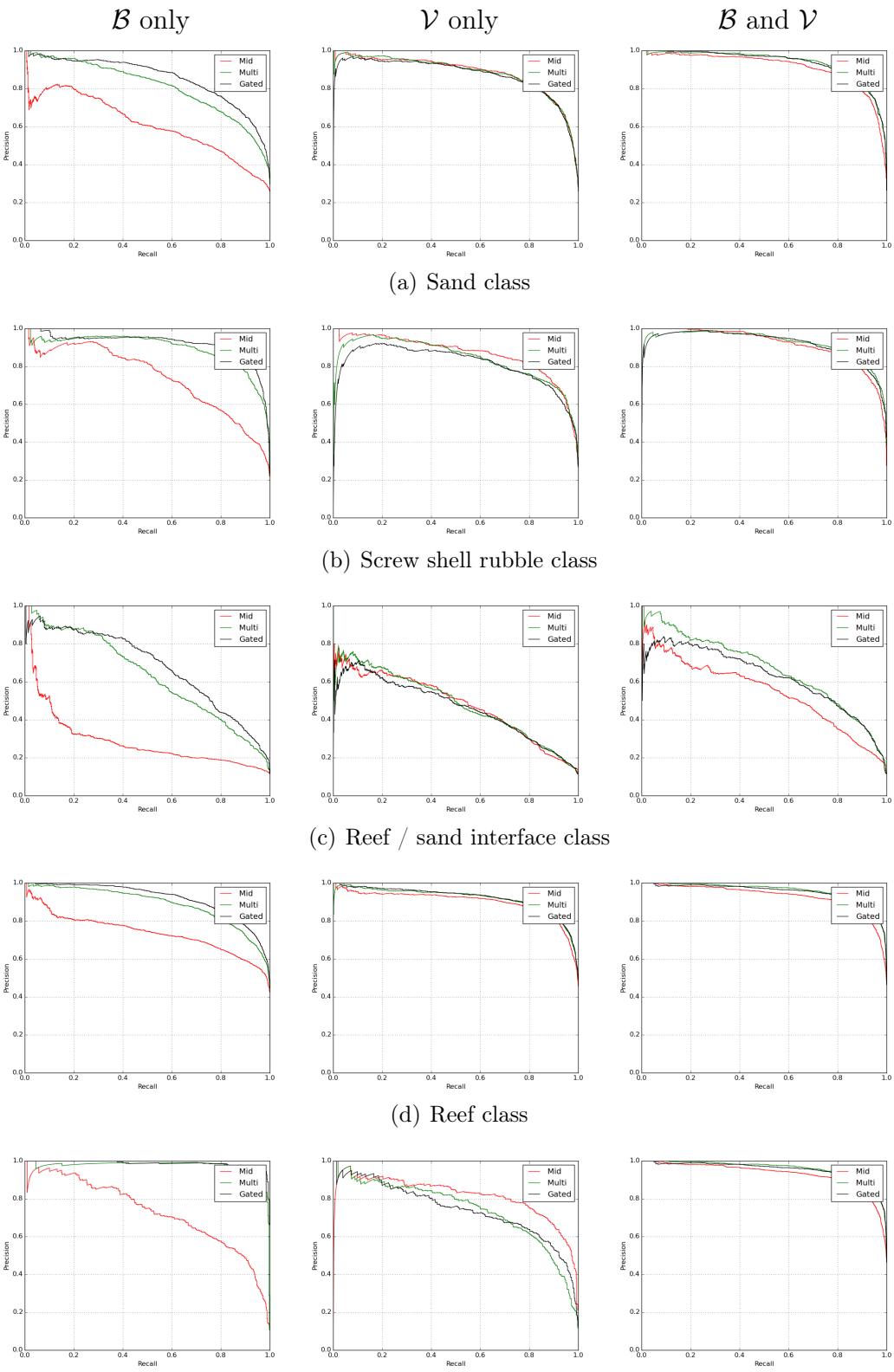
class, while the plots within each row each refer to a separate modality scenario (ie. which modalities are available): from left to right, they are the $\mathcal{B}$ only, $\mathcal{V}$ only, and $\mathcal{B}$ and $\mathcal{V}$ scenarios. Within each plot, the classifier for the gated shared layer feature encoding (black) is compared with the classifier for the standard shared layer feature encoding (green) and the midlayer feature encoding (red).

The precision-recall curves have similar characteristics to those in Chapter 4. Both multimodal encodings have a large effect when only bathymetry data is available (left hand column), but the gated model outperforms the standard model in this scenario. In contrast, it appears to perform on par with the standard multimodal model when visual data is available (middle) or both modalities are available (right hand). Interestingly, the gated model also performs more poorly for the kelp class when visual data is available.

5.5.4 Feature space analysis

Again, it is possible to better understand the effect of the gated model by analysing the gated shared layer feature representation using PCA. In this section, PCA is applied to the hidden features from all mixture components, extracted using the gated model.

Figure 5.5 shows the first four principal components of the gated layer features, with



(a) Sand class

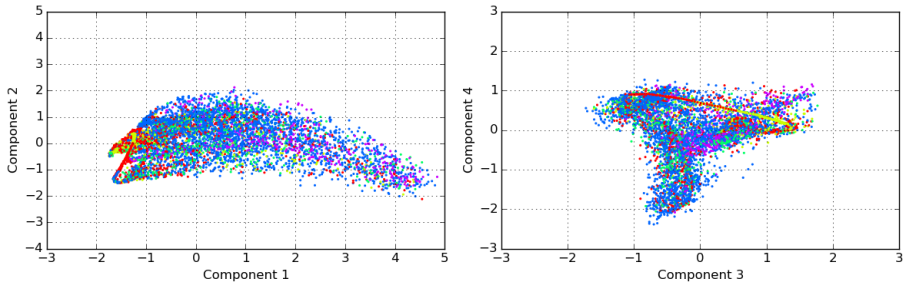
(b) Screw shell rubble class

(c) Reef / sand interface class

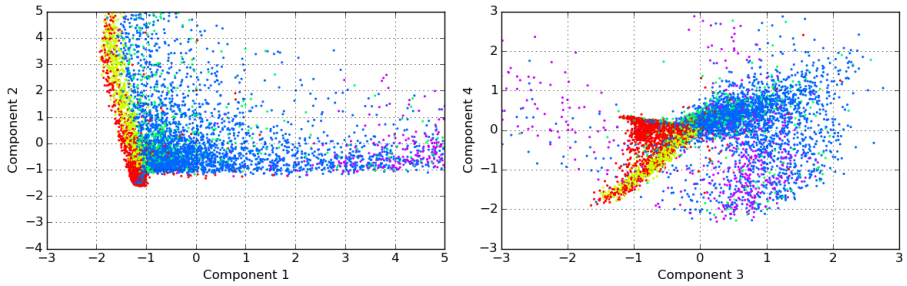
(d) Reef class

(e) Kelp class

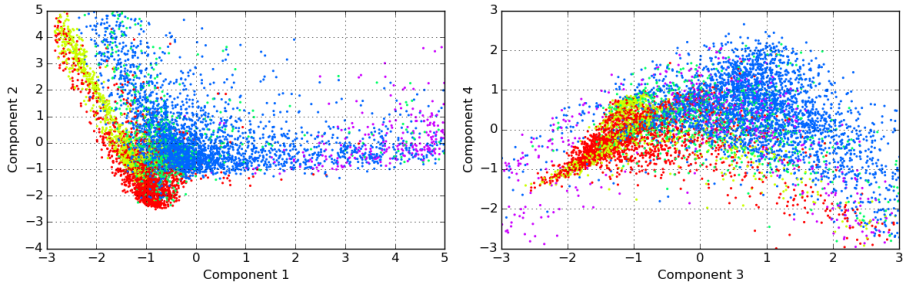
Figure 5.4 – Precision-recall curves for each habitat class, for each modality scenario. In each case, the left hand plot is for the bathymetry only ($\mathcal{B}$) scenario, the centre plot is for the visual only ($\mathcal{V}$) scenario, and the right hand plot is for both modalities ($\mathcal{B}$ and $\mathcal{V}$).



(a) Bathymetric features



(b) Visual features



(c) Both modalities

Figure 5.5 – The first four principal components for gated shared layer features

the first two principal components on the left-hand plot, and the third and fourth components on the right. Each row in the figure corresponds to a different modality scenario, with either bathymetric features ($\mathcal{B}$), visual features ($\mathcal{V}$), or both ($\mathcal{B} + \mathcal{V}$). The points used in each plot are coloured according to the corresponding habitat label.

It is interesting to note that the feature space of the gated shared layer looks remarkably different to the standard shared layer shown in Figure 4.4. Despite the fact that

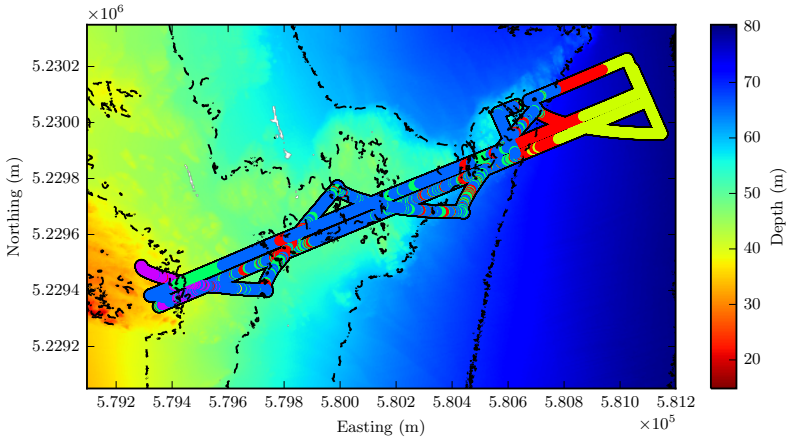
they yield very similar classification results, the feature mappings they learn are very different.

One noteworthy aspect is that the first principal component for bathymetric features correlates even more closely with the $\mathcal{V}$ and $\mathcal{B} + \mathcal{V}$ cases than in Figure 4.4. In particular, it is distributed over a very similar range, and has the same ordering of habitat classes across its range. This is again explained by the fact that the shared layer projects the inputs into the same high-dimensional space, such that the projected bathymetry data occupies the same feature space as the projected visual data.

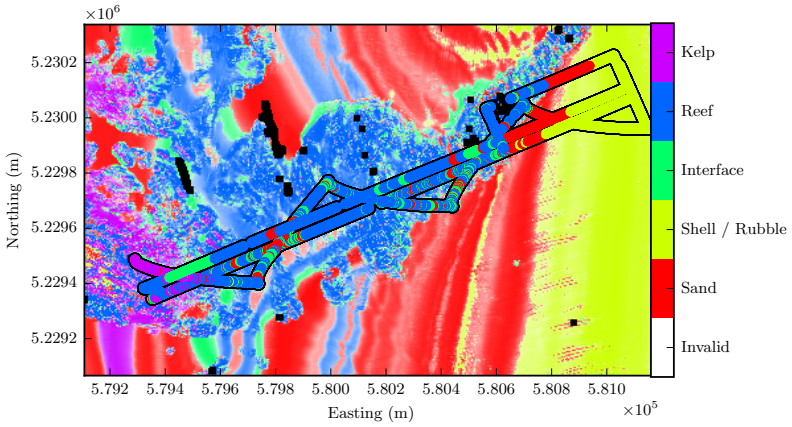
5.5.5 Habitat Mapping

As in Chapter 4, the multimodal encoded bathymetric features are used to perform habitat mapping in O'Hara Bluff. Figure 5.6(a) shows the bathymetry map for O'Hara, Figure 5.6(b) contains the habitat map produced, and Figure 5.6(c) shows the corresponding class probability maps. As with the previous experiments, the AUV trajectory is overlaid on the maps, coloured by the ground truth labels of the in-situ imagery, and the strength of the colour in the habitat map fades to white as the class probability is reduced.

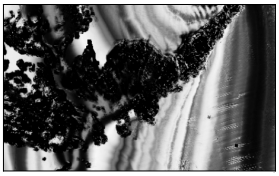
As with the multimodal model in Chapter 4, the habitat map is more fine-scaled than that with the midlayer features in Chapter 3. A few depth striations are still present, indicating a dependence on the depth value. However, the map also has some small improvements over Chapter 4; for example, the flat-bottomed shallow areas at the southwestern extent of the map are now classified as sand, which is more likely than the previous labels of reef and kelp. The map is also more expressive than that of Chapter 4: within the large contiguous expanse of reef, there are several patches containing the sand and interface classes.



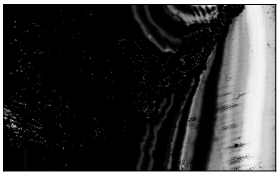
(a) Bathymetry map over the O'Hara Bluff region



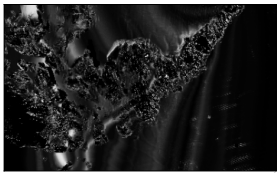
(b) Habitat map using gated layer features



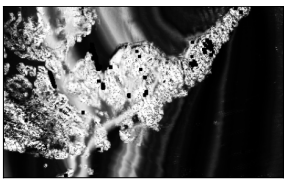
(i) Sand



(ii) Screw Shell Rubble



(iii) Reef / Sand Interface



(iv) Reef

(v) Kelp

(c) Individual class probability maps using gated layer features

Figure 5.6 – Habitat mapping results for the O'Hara Bluff region using gated layer features. Each map is overlaid with the habitat labels corresponding to images taken during AUV transects in the area. The classes are sand (red), screw shell rubble (yellow), reef / sand interface (green), reef (blue), and kelp (purple). The habitat map fades to white in uncertain locations. These images are best viewed in colour.

5.5.6 Clustering

Thus far, the performance of the gated multimodal model is very similar to the standard multimodal model. The remaining experiments demonstrate the ability of the gated model to perform additional unsupervised tasks that cannot be performed with the model from Chapter 4. One such task is the ability to perform unsupervised clustering of input data, which can be achieved by assigning each input vector to a single mixture component based on the associated mixture probabilities.

The gated model was applied to the multimodal data, and for each point, the most probable mixture component was determined (the assigned cluster). The 10 clusters with the greatest number of input samples are shown in Figure 5.7. It is important to note that the technique is clustering the data jointly over *both* visual and bathymetric inputs. Thus, while most images within each cluster are visually similar, some may be assigned according to bathymetric similarity.

As with the classification task, it is important to quantitatively evaluate the clustering performance for each modality scanerio: bathymetric data, visual data and both modalities. To do this, the cluster assignments by the algorithm can be compared with the ground truth class label, to see whether the class information can be extracted from the cluster assignment for each data point.

There are a number of clustering measures in the literature that accomplish this. The *homogeneity* evaluates whether each of the clusters only contain points that belong to a single class. As such, a value of 1 indicates that all of the points from every cluster belong to a single specific class. The *completeness* refers to the whether each class only contains points that are assigned to one cluster. i.e. a value of 1 indicates a perfect mapping from class label to cluster label. The *V-measure* is the harmonic mean of the homogeneity and completeness, which summarises the effect of both metrics. In fact, V-measure is also equivalent to the normalised mutual information (NMI), which is the mutual information between the the class labels and cluster labels, scaled by their individual entropies. Finally, the purity captures the fraction of total points that are correctly classified if each cluster is mapped to a single class based on its

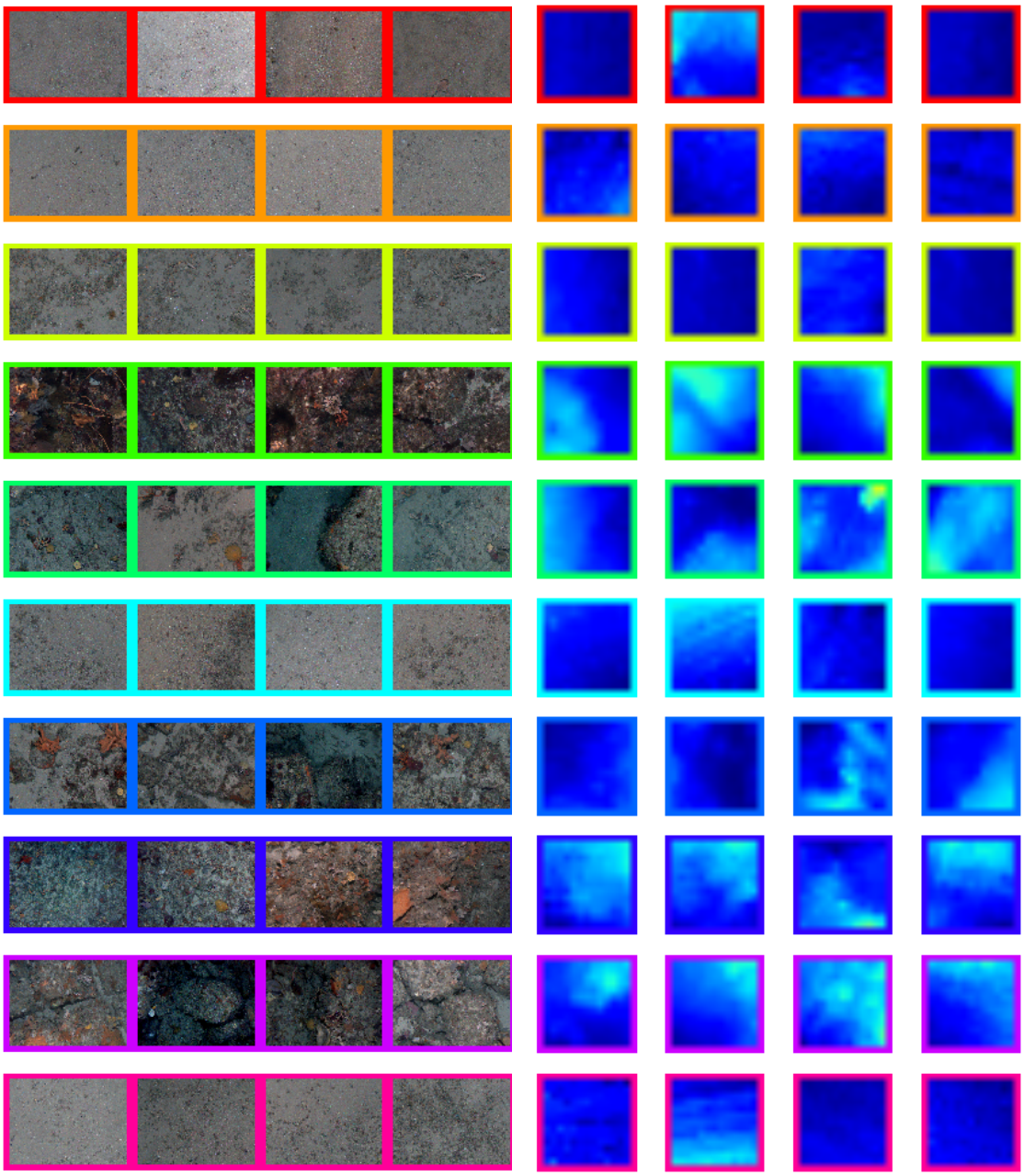


Figure 5.7 – Examples from the 10 largest clusters (each row). Each image (left) is matched with its corresponding bathymetric patch (right). Recall that the images typically have a footprint of approximately $2-3\text{m}^2$, while the bathymetric patches cover an area of 22.4m $\times$ 22.4m.

largest representing class.

The clustering results are shown in Table 5.2. The purity values are identical to

Table 5.2 – A number of clustering performance metrics for the different input modality scenarios.

	Modalities		
Metric	$\mathcal{B}$ and $\mathcal{V}$	$\mathcal{B}$ only	$\mathcal{V}$ only
Purity	0.778	0.616	0.715
Homogeneity	0.584	0.209	0.454
Completeness	0.278	0.198	0.296
V-measure	0.377	0.204	0.359

the $\mathbf{z}$ classification results in Section 5.5.2. This is to be expected, as by training a classifier on the cluster label $\mathbf{z}$, it learns to map the cluster to the most likely class. The results with both modalities and with visual data is very similar, and in fact, the completeness score and V-measure is lower for both modalities than for visual-only clustering. The lower completeness simply indicates that the model oversegments the class data into several clusters, which is acceptable for this application.

However, the bathymetry-only scenario does poorly compared to the other two. This suggests that, while the model can extract features from the bathymetry that perform well in classification, the cluster assignment itself is not very indicative of the underlying habitat. In other words, the habitat class for a bathymetric feature vector may be ambiguous if we only consider the most likely mixture component, but is usually clarified by considering the features from all components. This is further evidence for the one-to-many relationship outlined previously: with the bathymetric data there are many visual feature options, and selecting a single mixture component (mode) is not enough information for classification.

5.5.7 Visual prediction and image-based queries

With the ability to predict visual features in unseen areas, the gated model can additionally handle image-based queries, which can aid survey planning when supervised labels are not available. We present query-by-image results for the O'Hara Bluff, using the procedure in Section 5.4.3. Figure 5.8 shows query images from different

habitat classes and their resulting utility maps.

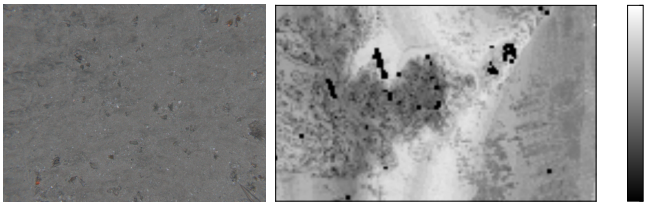
The results are visually similar to the class probability maps from Figure 5.6(c), and are consistent with the following known predictions. Sand images may be observed anywhere, but are more likely in the deep, flat-bottomed areas towards the East, while reef images are usually found in rugose (rugged terrain) regions. Images containing both sand and reef are likely to occur at the interface between the two, while kelp forests are restricted to shallower waters.

Interestingly, while there are still a few depth striations in the utility maps, they are weaker and fewer, as compared to the habitat probability maps from Figures 4.5(b) and 5.6(b). This indicates that the depth is a stronger feature for the supervised classification task than for the unsupervised learned relationship. That is, the supervised classifier utilises the depth value strongly, while the multimodal correlations learned in the unsupervised learning stage are distributed over a range of other bathymetric features as well.

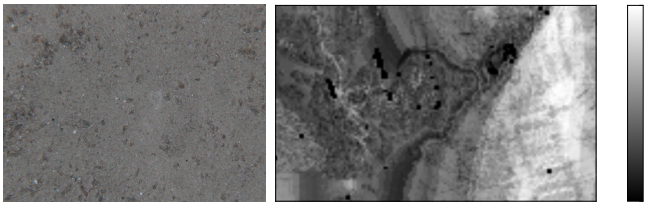
The results demonstrate that, without any supervision, the model can handle image-based queries and produce a utility map consistent with known class-based predictions.

5.6 Summary

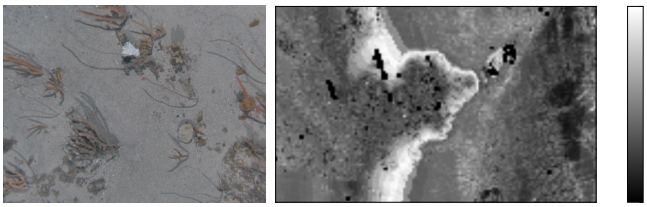
This chapter presented an alternative to the multimodal model outlined in Chapter 4, by using a gated mixture of RBMs model for the shared layer. This approach can better handle the one-to-many relationship between bathymetric and visual data, by using a gating variable to switch between different feature learners on-the-fly. A number of heuristics were proposed to avoid having to specify the number of components during training. Novel procedures were also introduced to compute the bathymetry-only mixture probabilities and the conditional expectations of visual features given bathymetric features. Together, these techniques allow the model to predict the different types of visual features that may be observed in previously unimaged areas



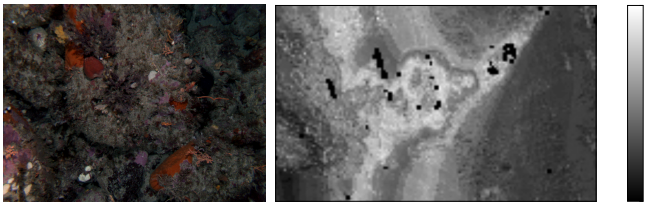
(a) Sand



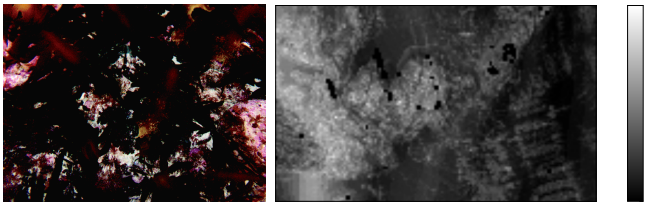
(b) Screw shell rubble



(c) Patch reef



(d) Reef



(e) Kelp

Figure 5.8 – Image-based query results for images from different habitat classes. Left: Query images. Right: Corresponding utility maps over the whole O'Hara Bluff region, normalised to span the range from black (lowest utility) to white (highest utility).

(and their associated probabilities), based on the bathymetry. This enables image-based queries, which can aid AUV survey planning, especially when supervised labels are unavailable.

Experiments were performed with a simulated dataset to demonstrate the benefits of the approach. Further experiments with the marine dataset demonstrated the effectiveness of the technique in classification, clustering, and visual prediction tasks.

Chapter 6

Information-theoretic measures for AUV survey planning

This chapter derives and discusses a number of information-theoretic measures to make use of multimodal data in AUV survey planning. The metrics are derived with respect to the MixRBM model detailed in the previous chapter. These measures are validated in two ways: they are used to generate utility maps over an entire region of interest, and the estimated utility is compared with actual benefit of each of the dives in the Southeastern Tasmania dataset.

This chapter is organised as follows. Section 6.1 provides some motivation for this problem and an overview of the approach. Section 6.2 provides a primer on information theory and explores some of the previous work on information-theoretic approaches for autonomous exploration. Section 6.3 presents the derivations of the proposed information-theoretic measures, and Section 6.4 validates these through experiments on both simulated and real marine data.

6.1 Overview

Due to the sheer size of the ocean environment, AUVs are unable to exhaustively sample the seafloor, and can only observe a tiny fraction of a larger region of in-

terest. This means that large-scale habitat classification is usually performed with bathymetry (ocean depth) data from shipborne multibeam Sonar, which is readily available prior to performing an AUV transect [14]. Since habitat classes are typically easier to distinguish in visual images, in-situ observation of habitats can help to resolve ambiguities in class labels and reduce uncertainty. Given the enormous area of interest, and the tight constraints involved with operating AUV dives in terms of resources, time, and cost, it is critical to select dive locations that optimise the visual information gained.

In this chapter, we propose a number of information-theoretic measures to predict the utility of unseen areas in terms of the expected visual information gain. These measures are designed to seek out locations where the expected visual data is likely to improve the certainty of the habitat map. Unlike other related approaches, the derived metrics are explicitly based on multimodal information: *both* the remotely sensed bathymetry data and the in-situ visual image observations.

We utilise the gated multimodal learning model from Chapter 5 to model the relationship between the two modalities and to predict visual image features from the bathymetric data. We then put forward novel derivations of two information-theoretic measures to aid survey planning. The approximations made in these derivations are justified through evaluation on a toy dataset. We also perform experiments with co-located bathymetry and visual image data, demonstrating that the proposed measures are strong indicators for the true utility of a dive, and that the resulting utility maps are consistent with scientific predictions.

6.2 A primer on information theory

Information theory is a field focused around quantifying the information content of data [58]. A central concept within the field is that of information *entropy*, which characterises the uncertainty in a random variable. For a random variable $\mathbf{y}$, the

entropy is given by:

$$\mathbb{H}(\mathbf{y}) = -\int_{\mathbf{y}} p(\mathbf{y}) \log p(\mathbf{y})\, d\mathbf{y} \tag{6.1}$$

The entropy is specified in *bits*, if the logarithm in Equation 6.1 has base 2, or in *nats*, if the natural logarithm is used [66]. In a sense, the entropy captures the amount of 'randomness' present in a variable: it reaches its maximum if $p(\mathbf{y})$ is a uniform distribution, and is minimised by a delta distribution with all of the probability mass attributed to a single value of $\mathbf{y}$.

If we have two random variables $\mathbf{x}$ and $\mathbf{y}$, we can quantify their relative information contents in a number of different ways. The conditional entropy $\mathbb{H}(\mathbf{y} \mid \mathbf{x})$ measures the uncertainty in the conditional distribution $p(\mathbf{y} \mid \mathbf{x})$. For the general case, where $\mathbf{x}$ is unobserved, it requires an expectation over all $\mathbf{x}$:

$$\mathbb{H}(\mathbf{y} \mid \mathbf{x}) = -\int_{\mathbf{x}} p(\mathbf{x}) \int_{\mathbf{y}} p(\mathbf{y} \mid \mathbf{x}) \log p(\mathbf{y} \mid \mathbf{x})\, d\mathbf{y} d\mathbf{x} \tag{6.2}$$

where observation of $\mathbf{x}$ reduces the corresponding integral to a single value of $\mathbf{x}$ with unit probability. This quantity represents the amount of information in $\mathbf{y}$ that cannot be "explained away" by observing $\mathbf{x}$.

A related quantity is the *mutual information*, denoted by $\mathbb{I}(\mathbf{x}; \mathbf{y})$, which quantifies the common information content of two variables. In other words, the mutual information predicts how much the observation of one variable tells us about the second.

$$\mathbb{I}(\mathbf{x}; \mathbf{y}) = \iint_{\mathbf{x},\mathbf{y}} p(\mathbf{x}, \mathbf{y}) \log \frac{p(\mathbf{x}, \mathbf{y})}{p(\mathbf{x})\, p(\mathbf{y})} d\mathbf{y} d\mathbf{x} \tag{6.3}$$

Using Jensen's inequality [66], we can derive the following bound on the mutual information:

$$\mathbb{I}(\mathbf{x}; \mathbf{y}) \quad = \quad -\iint_{\mathbf{x},\mathbf{y}} p(\mathbf{x}, \mathbf{y}) \log \frac{p(\mathbf{x})\, p(\mathbf{y})}{p(\mathbf{x}, \mathbf{y})} d\mathbf{y} d\mathbf{x}$$

$$\geq \quad -p(\mathbf{x},\mathbf{y}) \log \left(\iint_{\mathbf{x},\mathbf{y}} p(\mathbf{x},\mathbf{y}) \frac{p(\mathbf{x})\, p(\mathbf{y})}{p(\mathbf{x},\mathbf{y})} d\mathbf{y} d\mathbf{x} \right) = 0 \tag{6.4}$$

Equation 6.4 demonstrates that $\mathbb{I}(\mathbf{x};\mathbf{y}) \geq 0 \; \forall \, \mathbf{x}, \mathbf{y}$, and from Equation 6.3, we can see that the mutual information takes on a value of zero if and only if $p(\mathbf{x},\mathbf{y}) = p(\mathbf{x})\, p(\mathbf{y})$. i.e. the two variables are independent.

If we expand and rearrange Equation 6.3 using Bayes' rule, we obtain the following:

$$\begin{aligned}
\mathbb{I}(\mathbf{y} \mid \mathbf{x}) &= \iint_{\mathbf{x},\mathbf{y}} p(\mathbf{x},\mathbf{y}) \left[-\log p(\mathbf{y}) + \log \frac{p(\mathbf{x},\mathbf{y})}{p(\mathbf{x})} \right] d\mathbf{y} d\mathbf{x} \\
&= -\iint_{\mathbf{x},\mathbf{y}} p(\mathbf{x},\mathbf{y}) \left[\log p(\mathbf{y}) - \log p(\mathbf{y} \mid \mathbf{x}) \right] d\mathbf{y} d\mathbf{x} \\
&= -\int_{\mathbf{y}} p(\mathbf{y}) \log p(\mathbf{y})\, d\mathbf{y} + \int_{\mathbf{x}} p(\mathbf{x}) \int_{\mathbf{y}} p(\mathbf{y} \mid \mathbf{x}) \log p(\mathbf{y} \mid \mathbf{x})\, d\mathbf{y} d\mathbf{x} \\
&= \mathbb{H}(\mathbf{y}) - \mathbb{H}(\mathbf{y} \mid \mathbf{x})
\end{aligned} \tag{6.5}$$

The expression in Equation 6.5 is the difference between the entropy of variable $\mathbf{y}$ and the conditional entropy of $\mathbf{y}$ given $\mathbf{x}$. A similar derivation can show that the mutual information is also the difference between $\mathbb{H}(\mathbf{x})$ and $\mathbb{H}(\mathbf{x} \mid \mathbf{y})$. As a result, the mutual information can also be understood as the reduction in uncertainty of one variable after observation of the other [66], or the *expected information gained* by observing the second variable.

6.2.1 Application to autonomous exploration

This perspective of mutual information as a measure of expected information gain has led to its widespread use in robotic exploration applications. With such missions, the robot's goal is typically to explore an unseen environment and build a model of its surroundings, whether this is in the form of a metric map representation, or

semantic categorisation of objects and scenes that it encounters. In order to perform such exploration missions efficiently, the robot must be able to predict the expected information gain at each location. Trajectory planning can then be performed to ensure that an optimal amount of information is gathered during the mission.

A number of earlier works apply this to the Simultaneous Localisation and Mapping (SLAM) problem using LIDAR [12, 59, 87]. The goal is typically to select the control input in real time in order to balance the competing aims of localisation and exploration, aiming to minimise the pose uncertainty and the map certainty simultaneously. This is an interesting compromise, as the localisation accuracy usually has a pronounced effect on the accuracy of the resulting map as well. As such, the proposed techniques usually combine a number of information-theoretic metrics: the entropy of the robot's pose distribution, the expected information gain of a LIDAR scan, and in some cases, the cost of executing a particular control action as well [87]. By linearly combining these terms, the algorithm is able to find the balance between improving localisation accuracy and reducing uncertainty in the occupancy grid map.

Additionally, with the field of *active learning* gaining traction, information-based measures have been utilised to optimise classification performance with minimal labelling effort. In the traditional active learning paradigm, the model selects the most informative instances out of a set of unlabelled data, and queries the user for labels [33, 41]. As such, the model is able to produce a good classifier whilst simultaneously reducing the labelling burden.

One related work investigates active learning in a multi-class setting, using the entropy of the class probability distribution to identify samples to label [41]. By maximising the entropy measure, the model can seek out high uncertainty examples, for which it is most unsure about the class label. Another method aims to maximise the conditional mutual information between unlabelled instances and their corresponding labels, given the labelled data [33]. In this way, the model seeks out samples for which knowledge of the corresponding label will reduce the uncertainty of the remaining data.

This philosophy can be applied to the problem of selecting high utility survey locations

for AUVs. Rigby et al. [77] perform habitat modelling using a Gaussian Process (GP) classifier, and gauge the informativeness of a dive based on the predicted posterior entropy of the model (i.e. the remaining uncertainty after the observations have been made). They utilise Monte Carlo simulations to derive an upper and lower bound for the posterior entropy.

Another work proposes, in the absence of a prior model, that surveys should be placed in such a way that they observe as much of the bathymetric feature space as possible [6]. Accordingly, the utility of a candidate survey is based on the Kulback-Liebler divergence (KLD) between the feature distribution of the survey and that of the entire environment. This work is then extended in [7], using a GP to model the environment, and proposing a utility function based on expected information gain, which seeks out survey locations that minimise the variance of the habitat class predictions.

Girdhar et al. [30] utilise an online topic modelling algorithm to model the different types of terrain that may be encountered, and then perform trajectory planning based on word perplexity (confusion in the visual words that are observed) and topic perplexity (confusion in the topic labels). Their results demonstrate that their planning technique is able to find paths with high information content and the resulting "topic maps" closely match the ground truth.

These techniques demonstrate the ability to predict the utility of AUV survey locations, using measures such as the mutual information to assess the value of acquiring labelled imagery in unseen areas. However, the utility is based purely on a class label derived from the acquired image, and there may be benefit to more explicitly considering the additional visual features that are observed. We look to extend these techniques, utilising multimodal learning as a tool to capture the relationship between the bathymetry and visual data. The 'informativeness' of a location can be framed as the predicted information gain in terms of the visual features, rather than just the label information. This has the added benefit that new images do not have to be manually labelled in order to improve the model.

6.3 Information-theoretic measures for survey planning

In this section, we derive and discuss two multimodal information-theoretic measures to aid AUV survey planning. The metrics are derived with respect to the gated model detailed in Chapter 5.

The main goal of these measures is to select survey locations that maximise the amount of useful visual information acquired. Ideally, we would like to visit locations where the corresponding visual images are expected to significantly improve supervised habitat classification compared to the existing bathymetry data. Alternatively, if expert labels are unavailable, we would like the AUV to explore areas where the visual features are expected to hold a large amount of additional information given the bathymetry present.

6.3.1 Conditional mutual information

The gated multimodal model can generate features which can be used for large-scale habitat classification. The goal of the information-theoretic metric is then to determine, based on the bathymetry available, which locations to visit to improve the produced habitat map.

It may appear on initial consideration that a suitable measure would be the class uncertainty: the entropy of the class probabilities obtained by classifying the bathymetric features at each location. The problem with this measure is that it provides no indication about the value of additional visual information: visual observation may not yield any improvement if classes are poorly separated in *both* the visual and bathymetric feature spaces.

Instead, we want to predict how much *more* we expect to learn about the habitat label from a visual image, given that we already know the bathymetry. This quantity of interest is the Conditional mutual information (CMI), denoted by $\mathbb{I}(\mathbf{y}, \mathbf{x}_{\mathcal{V}} | \mathbf{x}_{\mathcal{B}})$: the

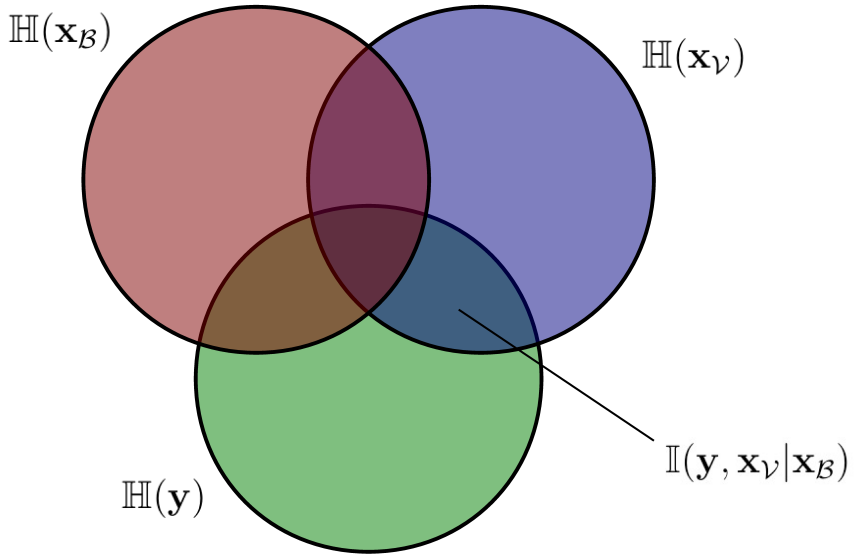


Figure 6.1 – Venn diagram showing the conditional mutual information term and its dependence on the entropies of the individual modalities.

expected value of the mutual information between labels $\mathbf{y}$ and visual features $\mathbf{x}_\mathcal{V}$ given bathymetry $\mathbf{x}_\mathcal{B}$. As the visual features $\mathbf{x}_\mathcal{V}$ are unobserved, this requires an integration over all $\mathbf{x}_\mathcal{V}$.

Figure 6.1 depicts this concept graphically. If each circle depicts the information (entropy) present in each of the three groups, we are interested in the area shown as $\mathbb{I}(\mathbf{y}, \mathbf{x}_\mathcal{V}|\mathbf{x}_\mathcal{B})$: the information common to visual features and labels that is absent from the bathymetry.

The CMI is given by the difference of two entropy terms:

$$\begin{aligned} \mathbb{I}(\mathbf{y}, \mathbf{x}_\mathcal{V}|\mathbf{x}_\mathcal{B}) &= \mathbb{H}(\mathbf{y} \mid \mathbf{x}_\mathcal{B}) - \mathbb{E}_{\mathbf{x}_\mathcal{V}} \left[\mathbb{H}(\mathbf{y} \mid \mathbf{x}_\mathcal{V}, \mathbf{x}_\mathcal{B})\right] \\ &= \mathbb{H}_\mathcal{B} - \mathbb{H}_{\mathcal{BV}} \end{aligned} \tag{6.6}$$

where $\mathbb{H}(\mathbf{y}|\cdot) = \sum_\mathbf{y} p(\mathbf{y} \mid \cdot) \log p(\mathbf{y} \mid \cdot)$ represents an entropy over the class probability distribution, and the shorthand terms $\mathbb{H}_{\mathcal{BV}}$ and $\mathbb{H}_\mathcal{B}$ refer to the class entropies with

and without visual information, respectively.

The term $\mathbb{H}_{\mathcal{BV}}$ in Equation 6.6 is the expected conditional entropy of the labels given both visual and bathymetric features:

$$\begin{aligned}\mathbb{H}_{\mathcal{BV}} &= \mathbb{E}_{\mathbf{x}_\mathcal{V}}\left[\mathbb{H}(\mathbf{y} \mid \mathbf{x}_\mathcal{V}, \mathbf{x}_\mathcal{B})\right] \\ &= -\int_{\mathbf{x}_\mathcal{V}} p\left(\mathbf{x}_\mathcal{V} \mid \mathbf{x}_\mathcal{B}\right) \mathbb{H}(\mathbf{y} \mid \mathbf{x}_\mathcal{V}, \mathbf{x}_\mathcal{B})\, d\mathbf{x}_\mathcal{V} \\ \mathbb{H}(\mathbf{y} \mid \mathbf{x}_\mathcal{V}, \mathbf{x}_\mathcal{B}) &= \sum_{\mathbf{y}} p\left(\mathbf{y} \mid \mathbf{x}_\mathcal{V}, \mathbf{x}_\mathcal{B}\right) \log p\left(\mathbf{y} \mid \mathbf{x}_\mathcal{V}, \mathbf{x}_\mathcal{B}\right)\end{aligned}$$

where the term $p\left(\mathbf{y} \mid \mathbf{x}_\mathcal{V}, \mathbf{x}_\mathcal{B}\right)$ can be computed by a classifier trained on the multimodal data. The sum over $\mathbf{y}$ is only over a small number of possible labels, but unfortunately, the integral over $\mathbf{x}_\mathcal{V}$ is intractable. Fortunately, our choice of model allows us to estimate this expectation using a discrete mixture-based approximation. We use each mixture component distribution in turn to find the conditional expectation of visual features given bathymetric features (denoted as $\mathbb{E}_k\left[\mathbf{x}_\mathcal{V} \mid \mathbf{x}_\mathcal{B}\right]$ for the k^{th} mixture component), and then use this small set of points to approximate the entire conditional distribution $p\left(\mathbf{x}_\mathcal{V} \mid \mathbf{x}_\mathcal{B}\right)$. This is equivalent to approximating a highly multimodal distribution by the set of points corresponding to the means of each of the modes, assuming that each mixture component models a single mode.

This approximation converts the computation into a tractable sum over mixture components:

$$\mathbb{H}_{\mathcal{BV}} \simeq -\sum_k p\left(z_k = 1 \mid \mathbf{x}_\mathcal{B}\right) \mathbb{H}(\mathbf{y} \mid \mathbb{E}_k\left[\mathbf{x}_\mathcal{V} \mid \mathbf{x}_\mathcal{B}\right], \mathbf{x}_\mathcal{B})$$

where $\mathbb{E}_k\left[\mathbf{x}_\mathcal{V} \mid \mathbf{x}_\mathcal{B}\right]$ is the conditional expectation of $\mathbf{x}_\mathcal{V}$ given $\mathbf{x}_\mathcal{B}$ under mixture component k.

The other term in Equation 6.6 is the conditional entropy of the labels given the bathymetry, and is given by:

$$\mathbb{H}_\mathcal{B} = -\sum_{\mathbf{y}} p\left(\mathbf{y} \mid \mathbf{x}_\mathcal{B}\right) \log p\left(\mathbf{y} \mid \mathbf{x}_\mathcal{B}\right)$$

We calculate $p(\mathbf{y} \mid \mathbf{x}_\mathcal{B})$ as follows, applying the same approximation as before.

$$\begin{aligned} p(\mathbf{y} \mid \mathbf{x}_\mathcal{B}) &= \int_{\mathbf{x}_\mathcal{V}} p(\mathbf{x}_\mathcal{V} \mid \mathbf{x}_\mathcal{B})\, p(\mathbf{y} \mid \mathbf{x}_\mathcal{V}, \mathbf{x}_\mathcal{B})\ d\mathbf{x}_\mathcal{V} \\ &\simeq \sum_k p(z_k = 1 \mid \mathbf{x}_\mathcal{B})\, p(\mathbf{y} \mid \mathbb{E}_k[\mathbf{x}_\mathcal{V} \mid \mathbf{x}_\mathcal{B}], \mathbf{x}_\mathcal{B}) \end{aligned}$$

While it may be tempting to model $p(\mathbf{y} \mid \mathbf{x}_\mathcal{B})$ more simply with a separate bathymetry-only classifier, this can be problematic: with two different classifiers, there is no guarantee that the class probabilities they assign will be consistent with one another. For example, the bathymetry-only classifier may underestimate its uncertainty while the distribution $p(\mathbf{y} \mid \mathbb{E}_k[\mathbf{x}_\mathcal{V} \mid \mathbf{x}_\mathcal{B}], \mathbf{x}_\mathcal{B})$ under a different classifier is more realistic and has higher entropy. Experiments have shown that using two separate classifiers in this computation can yield inconsistent results such as negative mutual information.

6.3.2 Conditional entropy

It is also desirable to seek out regions in which the visual data is expected to hold a lot of information, independent of class labels. This ensures that a planning metric is available even when expert labels are not.

For this measure, the quantity of interest is the Conditional entropy (CE), denoted by $\mathbb{H}(\mathbf{x}_\mathcal{V} \mid \mathbf{x}_\mathcal{B})$. A large value for the CE at a particular location indicates that the bathymetric features convey very little information about the visual features.

$$\mathbb{H}(\mathbf{x}_\mathcal{V} \mid \mathbf{x}_\mathcal{B}) = -\int_{\mathbf{x}_\mathcal{V}} p(\mathbf{x}_\mathcal{V} \mid \mathbf{x}_\mathcal{B}) \log p(\mathbf{x}_\mathcal{V} \mid \mathbf{x}_\mathcal{B})\ d\mathbf{x}_\mathcal{V} \tag{6.7}$$

This has the same intractable sum over $\mathbf{x}_\mathcal{V}$, but using the same mixture-based approximation applied previously, we find that:

$$\mathbb{H}(\mathbf{x}_\mathcal{V} \mid \mathbf{x}_\mathcal{B}) \simeq -\sum_k p(z_k = 1 \mid \mathbf{x}_\mathcal{B}) \log p(z_k = 1 \mid \mathbf{x}_\mathcal{B}) \tag{6.8}$$

In other words, we approximate the conditional entropy of visual features given bathymetry by the entropy of the bathymetry-only mixture probabilities $p\left(z_k = 1 \mid \mathbf{x}_\mathcal{B}\right)$.

6.4 Experiments

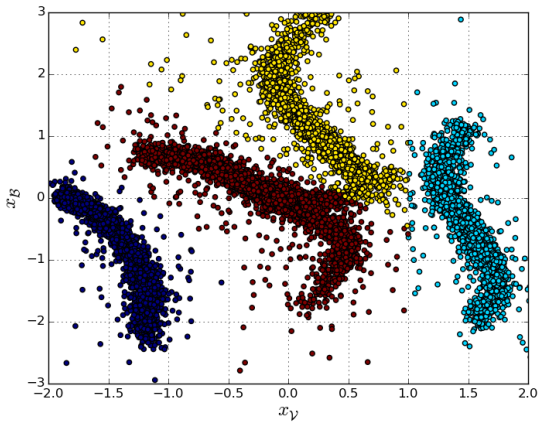
For this chapter, experiments are again conducted with both simulated and real data, as in Chapter 5. The gated multimodal model trained in Chapter 5 is used for all experiments.

6.4.1 Toy results

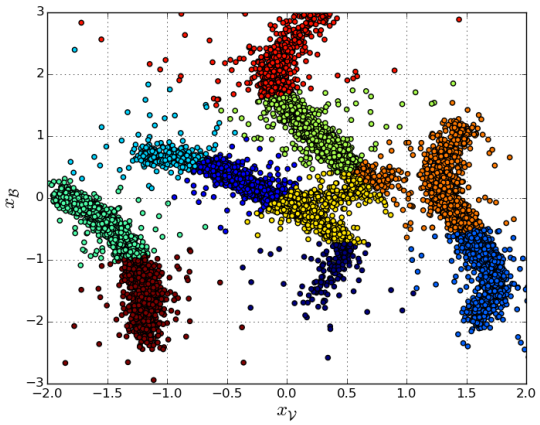
To better understand the effects of the two metrics, and to validate the approximations made in their derivations, we evaluate them on the toy dataset introduced in Chapter 5 For this toy dataset, the equivalent "mission planning" task is to select a setpoint of $x_\mathcal{B}$ (a 'slice' of the input space) where the visual feature dimension $x_\mathcal{V}$ is likely to yield the most useful information. In this case, the CMI measure should suggest selecting an $x_\mathcal{B}$ such that knowledge of $x_\mathcal{V}$ would be *most* useful in determining the class label. In contrast, the CE metric simply suggests an $x_\mathcal{B}$ where there is greatest corresponding variation / entropy in $x_\mathcal{V}$.

In the CMI plot in Figure 6.2(e), we observe two main peaks, at approximately -1.5 and 0.9. At both of these setpoints, there are three potential classes that are clearly separable if the $x_\mathcal{V}$ value is known. The CMI is lower at $x_\mathcal{B} = 0$, where the classes have more overlap, and is zero at $x_\mathcal{B} > 2$ when the yellow class dominates (Figure 6.2(c)).

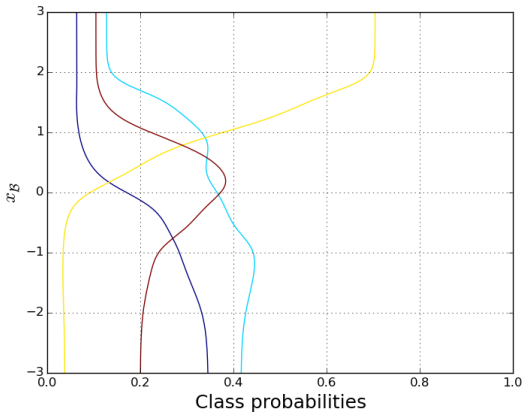
The CE plot exhibits a similar behaviour, but is related to the number of mixture components for a given $x_\mathcal{B}$ instead of the number of classes. Figure 6.2(b) demonstrates how the model utilises different mixture components for different parts of the input space, and Figure 6.2(d) shows the model's estimate for the marginal mixture probabilities, as a function of $x_\mathcal{B}$ alone. In line with these plots, the CE measure (Figure 6.2(f)) is high for $-1 < x_\mathcal{B} < 1$, where there are up to six mixture components



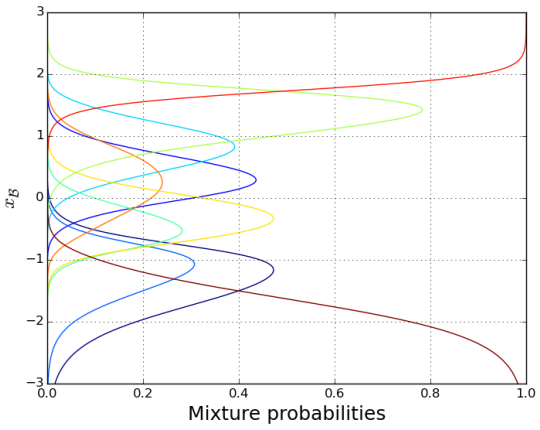
(a) Toy dataset coloured by class label



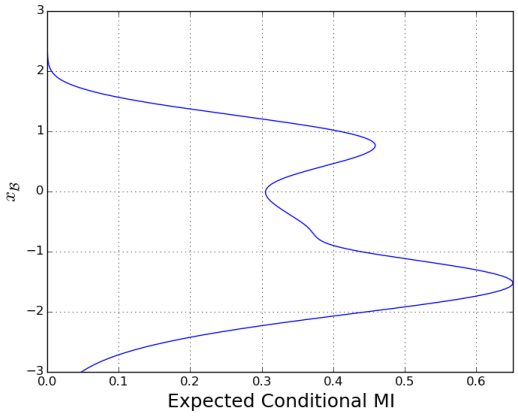
(b) Toy dataset coloured by mixture component



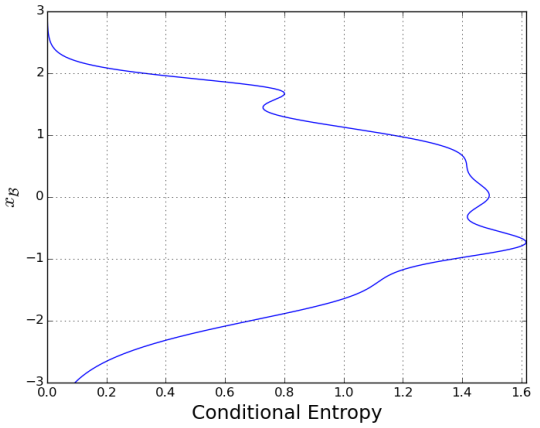
(c) Conditional class probabilities for each bathymetric value



(d) Conditional cluster probabilities for each bathymetric value



(e) Conditional Mutual Information for each bathymetric value



(f) Conditional entropy for each bathymetric value

Figure 6.2 – Experimental results demonstrating the information-theoretic metrics on the 2D toy dataset. The dataset is designed such that the conditional distribution $p(x_{\mathcal{V}}|x_{\mathcal{B}})$ can be highly multimodal.

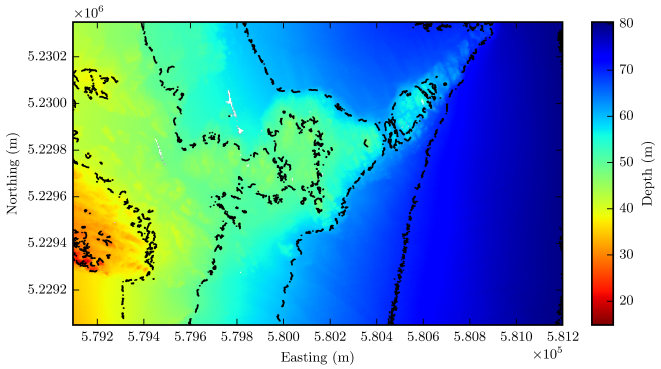
used, and close to zero at very low or high $x_{\mathcal{B}}$, where the conditional distribution has only a single mode. At unimodal locations, there is very little benefit to observing $x_{\mathcal{V}}$, compared to a location where $x_{\mathcal{V}}$ can take on several different values.

Thus, the toy results demonstrate the benefits of the derived information-theoretic measures, and justify the approximations made in the derivation process. Both metrics are able to select the values of $x_{\mathcal{B}}$ such that subsequent observation of $x_{\mathcal{V}}$ is most useful. This is a direct analogue for the real-world application, of finding locations (based on the bathymetry) where the observed visual images are expected to yield the greatest information gain.

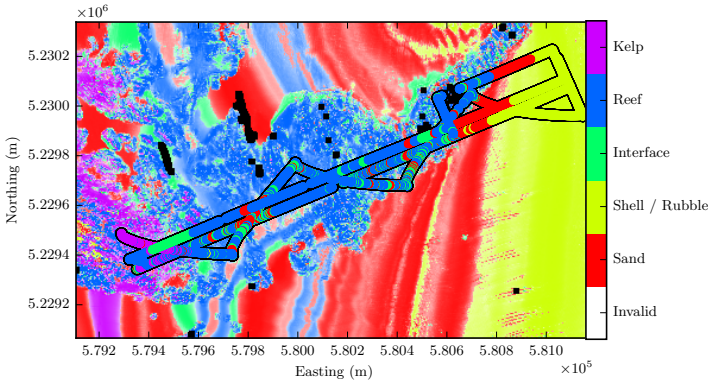
6.4.2 Predictive utility mapping

The key benefit of the derived measures is that they can aid survey planning, by indicating locations that are likely to yield high reward within a larger region. In this section, we use the model to calculate the CMI and CE over a region in Southeastern Tasmania known as O'Hara Bluff. Figure 6.3(a) shows the bathymetry over the region, while Figure 6.3(b) shows the habitat map generated by classifying the features extracted by the multimodal learning model. The CMI and CE maps are shown in Figures 6.3(c) and 6.3(d).

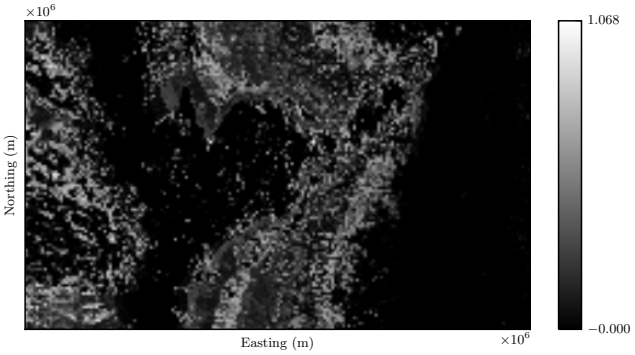
In general, a key habitat indicator is the *rugosity*, or terrain roughness, of the local bathymetry. Highly rugose regions are very likely to be reef or kelp, while flat-bottomed areas are predominantly sand or rubble. Both the CMI and CE maps are consistent with this prior knowledge. The deeper, flatter regions towards the east are almost certainly screw shell rubble, and there is little value in observing these areas. In a similar fashion, the rugose areas at moderate depths ($40 - 60$m in Figure 6.3(a)) are very likely to be reef, and the CMI and CE measures assign low utility to these regions. Since kelp is usually only found in shallower waters, there is greater value in exploring the shallow region towards the west, where visual information can distinguish between the reef and kelp classes. Another region of ambiguity, in terms of the known bathymetry, is in the interface between the rugose



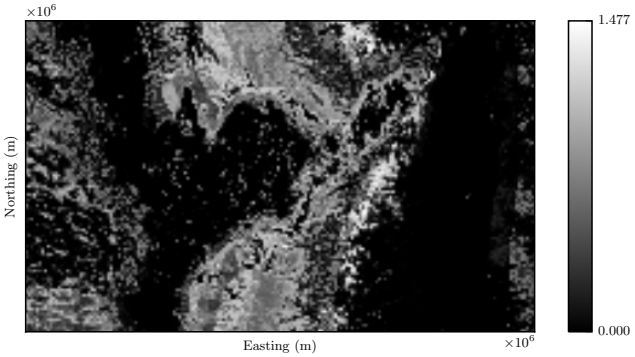
(a) Bathymetry for O'Hara Bluff region



(b) Habitat map generated under the gated multimodal model



(c) Conditional mutual information map



(d) Conditional entropy map

Figure 6.3 – Information-theoretic utility maps generated for the O'Hara Bluff region. The habitat map is produced by classifying the hidden features of the mixture of RBMs model.

reef habitats and the flat-bottomed sand or rubble areas, since different types of visual features and habitat classes may be observed.

One subtle difference between the CMI and CE maps is that the CE utility appears to strongly prefer the interface region, while the CMI assigns similar utility to the interface region and shallow areas. This is because the CMI explicitly takes into account the discriminability of the different habitat classes, while the CE only looks at the entropy or uncertainty of the visual features. This suggests that while the interface region has a high entropy in terms of the different visual features that may be observed, it is equally beneficial to survey the shallower areas in order to resolve habitat class ambiguities.

As a result, the CMI and CE measures provide introspective capabilities for the multimodal learning model. They are able to predict the uncertainty in unobserved regions, and are in agreement with expert predictions. In particular, by utilising *multimodal* information-theoretic measures, we are able to predict the regions which are expected to provide the greatest visual information gain, given the bathymetric information already available.

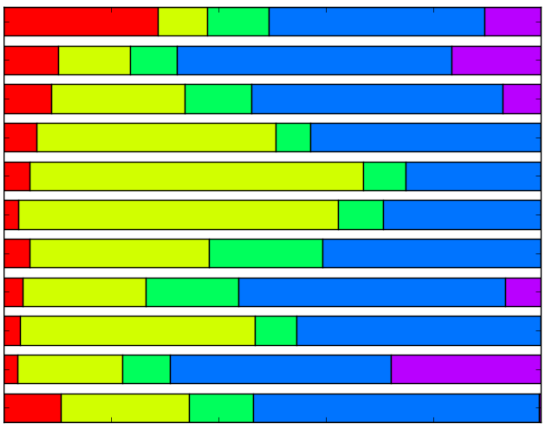
6.4.3 Survey selection

Ultimately, the proposed measures must be able to predict the locations for which visual observation is likely to yield greatest improvement in performance. For areas assigned high utility, we would expect that the inclusion of visual information would increase the probability of selecting the correct habitat class.

We analyse this effect quantitatively using the entire Southeastern Tasmania dataset. Table 6.1 shows the distribution of labels for each dive within this dataset, along with the entropy of this label distribution.

For this experiment, we first divide the dataset evenly into a training set and a test set, and train the multimodal model on the training data. We also train two classifiers on the training points: one using just the bathymetric data, and one using the multimodal features; and we apply these to the test set to obtain the class probabilities

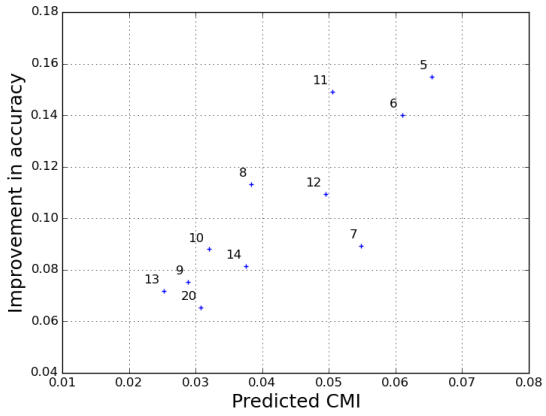
Table 6.1 – The distribution of habitat labels over each dive, summarised by the entropy value. The classes are sand (red), screw shell rubble (yellow), sand / reef interface (green), reef (blue), and kelp (purple).

Dive	# images	Habitat distribution	Entropy
5	11361		1.43
6	6459		1.36
7	10818		1.36
8	6138		1.07
9	5658		0.99
10	5819		0.97
11	6525		1.21
12	5325		1.29
13	5311		1.03
14	5903		1.35
20	6110		1.19

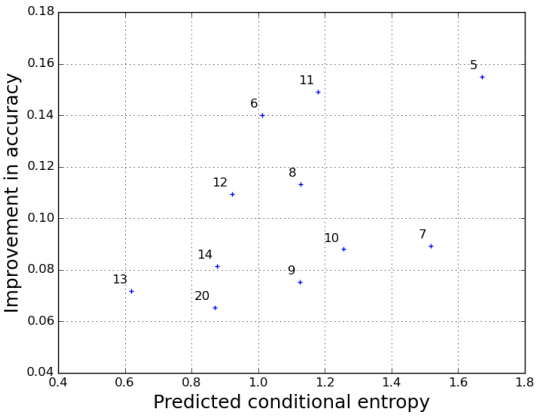
at each location. Given the labels for each test point, we compute the increase in the probability of the correct class when using the multimodal model versus the base bathymetric model. This metric indicates the true utility of the dive, as it measures the actual effect of incorporating visual information with respect to the true class label. We compare this value with the predicted CMI and CE.

To summarise the analysis, we report the mean values over each dive in the dataset. The quantitative results are shown in Figure 6.4. In Figure 6.4(a) we observe a correlation between the average CMI over a dive, and the mean increase in accuracy by incorporating visual features for the dive. Figure 6.4(b) demonstrates a similar relationship for the CE measure. The Spearman rank coefficients for the two plots are 0.88 and 0.71 respectively, indicating that the measures can be used to rank a set of a candidate surveys based on their expected utility.

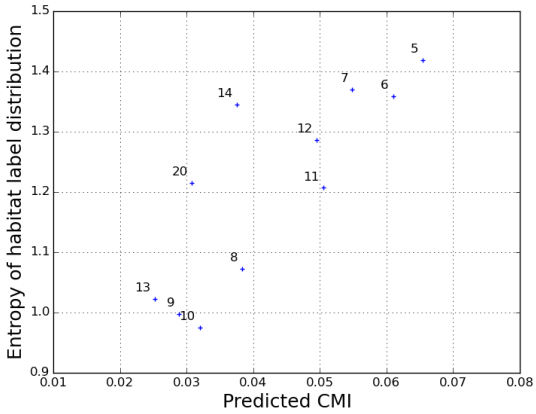
Interestingly, if we look at the distribution of true habitat labels (Table 6.1), the CMI also appears to be correlated with the entropy of this distribution (Figure 6.4(c)), with Spearman rank coefficient of 0.79. As a result, the CMI may also act as a good indicator of the survey locations that are likely to cover a wide range of habitats.



(a) CMI versus mean increase in accuracy



(b) CE versus mean increase in accuracy



(c) CMI versus entropy of habitat label distribution

Figure 6.4 – Predicted utility versus true utility for each dive in the SE Tasmania dataset. True utility can be measured in terms of the improvement in classification performance or the spread (entropy) of true habitat labels from the survey.

6.5 Summary

In this chapter, a number of multimodal information-theoretic measures were proposed to aid survey planning for AUVs. The gated multimodal learning approach from Chapter 5 was used to capture the relationship between remotely sensed bathymetry data and in-situ visual observations. By using the model to predict visual features from bathymetric data, it is possible to predict the utility of unobserved areas, in terms of the expected additional information gained by visual observation. Results on a 2D toy dataset suggest that the approximations made by the model are reasonable, while experiments on the Southeastern Tasmania data demonstrate the ability to predict the informativeness of a survey location.

Chapter 7

Conclusions

The purpose of this thesis is to investigate multimodal learning techniques from visual and remotely sensed data, applied to the problem of autonomous exploration and monitoring with AUVs.

AUVs are able to obtain very large quantities of visual imagery through in-situ observation of the ocean floor. However, since they are only able to traverse a tiny fraction of the ocean floor, remotely sensed bathymetry data from shipborne multibeam sonar is necessary to perform large-scale habitat classification. Nonetheless, visual observation of the seafloor can resolve ambiguities in habitat predictions. It is important to leverage the benefits of these modalities when performing classification tasks.

Multimodal learning addresses this goal. By modelling the relationship between the two modalities, it is possible to achieve improved classification accuracy, as well as enable additional inference tasks that can aid survey planning. Further, such a model facilitates information-theoretic measures for survey planning that predict the amount of useful visual information in unobserved areas, as a function of the known bathymetry data.

The primary contributions of this thesis are summarised in the following section.

7.1 Contributions

Four main contributions are presented in this thesis, as detailed below.

7.1.1 Feature learning from marine data

Chapter 3 describes a novel application of feature learning techniques to marine data, for both the visual images and bathymetry.

The features learned from the bathymetric data are compared with the features that are traditionally used for habitat classification: rugosity, slope, and aspect. Experiments demonstrate that the learned features capture the important rugosity, slope and aspect information, and perform better in classification tasks.

The visual feature learning technique proposed by Steinberg [89] is compared with a number of CNN architectures, since CNNs achieve state-of-the-art performance in a range of computer vision and machine learning tasks. Experiments demonstrate that the approach captures high-level factors of variation in the data, and performs similarly well to the CNN architectures.

7.1.2 Multimodal learning from visual and bathymetric data

The learned features for both modalities are then utilised in a multimodal model in Chapter 4, which captures the correlations between the two modalities. To the best of our knowledge, this represents the first use of multimodal learning for AUV applications.

Experiments are performed with co-located visual and bathymetry data, and demonstrate improved classification performance, regardless of which modalities are available. The key benefit is that, by providing both modalities at feature learning time, the model learns better features for each modality individually, which is beneficial if only one modality is available at classification time. As a result, the model can more

accurately perform large-scale benthic habitat mapping, where only the bathymetric data is available.

This is a novel way of framing the traditional habitat mapping problem. Rather than the classification of purely bathymetric features, this approach considers the task as one of joint learning on bathymetry and image data with only one of the modalities available for large-scale inference.

7.1.3 Gated models for multimodal learning

Chapter 5 then proposes an extension to the standard multimodal learning paradigm: the use of a gated model in the multimodal layer. This model is able to learn multiple RBM components under the same framework, which can better capture the one-to-many relationship that exists between the bathymetry and the visual features. A number of extensions are proposed to the gated model, including heuristics to automatically determine the number of cluster components k during training, and algorithms to predict k sets of visual features from bathymetric data, each with an associated mixture probability.

As demonstrated by experiments on simulated data and real marine data, the model achieves very similar classification accuracy to the model proposed in Chapter 4. Additionally, the ability to predict visual features from bathymetric data affords the option of handling image-based queries, where the model can determine areas in which an input image is likely to be observed. Such queries are very useful in survey planning, particularly in scenarios where expert habitat labels are unavailable.

7.1.4 Information-theoretic measures for survey selection

The final contribution of this thesis is to derive information-theoretic measures to predict the expected utility of unobserved areas (Chapter 6). Unlike previous work, the proposed measures are explicitly multimodal metrics: they predict the expected information gained by in-situ visual observation, given the known bathymetry data.

As such, the utility is based on the informativeness of the entire set of image features, rather than just the observed label.

Experiments with simulated data suggest that the approximations made in the derivation are sound. The measures are then applied to large-scale bathymetry, and the resulting utility maps are consistent with scientific predictions. Finally, experiments over the whole Southeastern Tasmania dataset demonstrate that the measures correlate well with the improvement in classification accuracy by observing an image at each location, and also tend to select dives which cover a range of habitats.

7.2 Future Work

7.2.1 Multimodal learning for autonomous ground vehicles

This thesis has investigated multimodal learning from visual and remotely sensed data, focusing on the use of AUVs in exploration and monitoring tasks. However, similar algorithms could be useful in autonomous ground vehicles in urban environments. 3D point cloud data from a laser range scanner has some similar characteristics to bathymetric data: it is coarser than visual information as it has lower spatial resolution, but can provide information on topological structure and shape.

By learning the relationship between visual image data and laser scan information, a multimodal model would be able to perform similar inference tasks to those proposed in this thesis: improving semantic classification performance from laser, and predicting visual features. Given that laser scans often have a finer spatial resolution than acoustic bathymetry grids, multimodal learning may also lead to improved inference at the sub-image level, such as pixel-wise or segment-wise classification.

7.2.2 Incorporation of acoustic backscatter data and other modalities

While shipborne multibeam sonar is able to provide bathymetry data through time-of-flight ranging, it can also provide *backscatter* data through the intensity of the return. The backscatter can also provide useful information about the benthic habitat, as it is, in part, a function of the absorptive properties of the seafloor.

Unfortunately, the backscatter maps produced can be highly susceptible to noise, and to a number of artefacts, such as nadir effects and outer beam artefacts. This means that a large amount of post-processing is required to utilise the data. Future work will look at machine learning and computer vision models to address these issues, and incorporate the information into the multimodal learning process.

Further, other available modalities could be used for learning, including bathymetry or backscatter from an AUV-mounted sonar, or dense seafloor reconstructions from the onboard stereo cameras.

7.2.3 Information-theoretic trajectory planning

This thesis has proposed information-theoretic measures to aid survey planning, by predicting the expected utility of visual information given the available bathymetric data. Future work will look to build on this by integrating the metrics more directly into a trajectory planning algorithm. Crucially, such algorithms would seek to tradeoff between spatial exploration of the seafloor and exploitation of the existing model.

As a first pass, the measures can be used to generate a utility map, and the trajectory planning problem can be posed as a Travelling Salesman Problem or Coverage Salesman Problem, aiming to visit all of the locations with high predictive utility. Alternatively, the measures could be used in a reward function under a reinforcement learning or Partially Observable Markov Decision Process (POMDP) framework.

7.2.4 Improved training of gated models

The gated model used in Chapter 5 for the multimodal layer can be interpreted in a different way: rather than a mixture of k RBMs, it can be considered as a single RBM with its hidden units partitioned into k equal blocks, with the added freedom of utilising a different set of visible biases for each block. From this perspective, using the gated model versus a standard RBM is arguably similar to imposing a strict sparsity constraint during training, such that only units in one of the k groups can be on for any given input vector. While the gated model provides a number of additional benefits over a standard RBM (as outlined in this thesis), removing this 'hard constraint' during training could lead to even better performance.

An alternative approach could be to commence training the model as a standard RBM, and monitor the hidden activations over the entire dataset. If, at any point during training, the hidden activations can be naturally partitioned into different groups, the hidden units can be split into the different mixture components, and training would proceed as for a gated model. The additional benefit here would be that each mixture component could be assigned different numbers of hidden units as necessary. Future work will investigate this possibility.

7.2.5 Experimental validation across multiple environments

While the proposed models have been extensively evaluated on the entire southeastern Tasmania dataset, an interest direction would be to investigate their efficacy on other marine environments and habitat classes. For example, the benthic habitats found in tropical waters are likely to be vastly different to the temperate waters of southeastern Tasmania, both in terms of their visual appearance and the bathymetric variables defining the seafloor topography. Ideally, a model trained on one environment could be adapted to another in an online learning framework, such that an existing model can still act as a prior for the multimodal relationship in a new environment.

Bibliography

[1] Guillaume Alain and Yoshua Bengio. What regularized auto-encoders learn from the data-generating distribution. *The Journal of Machine Learning Research*, 15(1):3563–3593, 2014.

[2] Timothy J Alexander, Neville Barrett, Malcolm Haddon, Graham Edgar, and Others. Relationships between mobile macroinvertebrates and reef structure in a temperate marine reserve. *Mar Ecol Prog Ser*, 389:31–44, 2009.

[3] Richard G Baraniuk and Michael B Wakin. Random projections of smooth manifolds. *Foundations of computational mathematics*, 9(1):51–77, 2009.

[4] Oscar Beijbom, Peter J Edmunds, David Kline, B Greg Mitchell, David Kriegman, and Others. Automated annotation of coral reef survey images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1170–1177, 2012.

[5] Asher Bender, Stefan B Williams, and Oscar Pizarro. Classification with Probabilistic Targets. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 1780–1786, 2012.

[6] Asher Bender, Stefan B. Williams, and Oscar Pizarro. Autonomous exploration of large-scale benthic environments. *IEEE International Conference on Robotics and Automation*, pages 390–396, 2013.

[7] Asher Bender, Stefan B Williams, and Oscar Pizarro. Autonomous Methods for Environmental Modelling and Exploration. In *Robotic Science and Systems (RSS) workshop on Robotic Exploration, Monitoring, and Information Collection: Nonparametric Modeling, Information-based Control, and Planning under Uncertainty*, 2013.

[8] Yoshua Bengio, Li Yao, Guillaume Alain, and Pascal Vincent. Generalized denoising auto-encoders as generative models. In *Advances in Neural Information Processing Systems*, pages 899–907, 2013.

[9] C M Bishop. *Pattern recognition and machine learning*. Springer, 2006.

[10] Liefeng Bo. Hierarchical Matching Pursuit for Image Classification: Architecture and Fast Algorithms. In *Neural Information Processing Systems (NIPS)*, pages 2115–2123, 2011.

[11] Liefeng Bo, Xiaofeng Ren, and Dieter Fox. Unsupervised feature learning for RGB-D based object recognition. In *13th International Symposium on Experimental Robotics*, pages 1–15, 2012.

[12] Frédéric Bourgault, Alexei Makarenko, Stefan B Williams, Ben Grocholsky, Hugh F Durrant-Whyte, and Others. Information based adaptive robotic exploration. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 1, pages 540–545, 2002.

[13] C J Brown and R Coggan. Verification of acoustic classes. *Acoustic seabed classification of marine physical and biological landscapes. ICES Co-operative Research Report*, 286:127–144, 2007.

[14] Craig J. Brown, Stephen J. Smith, Peter Lawton, and John T. Anderson. Benthic habitat mapping: A review of progress towards improved understanding of the spatial ecology of the seafloor using acoustic techniques. *Estuarine, Coastal and Shelf Science*, 92(3):502–520, 2011.

[15] Rozaimi Che Hasan, Daniel Ierodiaconou, Laurie Laurenson, and Alexandre Schimel. Integrating multibeam backscatter angular response, mosaic and bathymetry data for benthic habitat mapping. *PloS one*, 9(5):e97339, 2014.

[16] Brian Cheung, Jesse a. Livezey, Arjun K. Bansal, and Bruno a. Olshausen. Discovering Hidden Factors of Variation in Deep Networks. In *arXiv preprint arXiv:1412.6583*, page 12, 2014.

[17] Adam Coates, Honglak Lee, and Andrew Y AY Ng. An analysis of single-layer networks in unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, pages 1–9, 2010.

[18] Adam Coates, Andrew Y Ng, and Serra Mall. The Importance of Encoding Versus Training with Sparse Coding and Vector Quantization. In *International Conference on Machine Learning*, page 10, 2011.

[19] Aaron Courville. A spike and slab restricted Boltzmann machine. In *International Conference on Artificial Intelligence and Statistics*, volume 15, pages 233–241, 2011.

[20] Alexey Dosovitskiy and Thomas Brox. Inverting convolutional networks with convolutional networks. *arXiv preprint arXiv:1506.02753*, pages 1–15, 2015.

[21] Bertrand Douillard, Dieter Fox, F Ramos, and H Durrant-Whyte. Classification and semantic mapping of urban environments. *The International Journal of Robotics Research*, 30(1):5–32, 2011.

[22] Matthew Dunbabin and Lino Marques. Robots for environmental monitoring: Significant advancements and applications. *Robotics & Automation Magazine*, 19(1):24–39, 2012.

[23] Dumitru Erhan, Yoshua Bengio, Aaron Courville, and Pascal Vincent. Visualizing Higher-Layer Features of a Deep Network. Technical report, 2009.

[24] Dumitru Erhan, Yoshua Bengio, Aaron Courville, Pierre-Antoine Manzagol, Pascal Vincent, and Samy Bengio. Why does unsupervised pre-training help deep learning? *The Journal of Machine Learning Research*, 11:625–660, 2010.

[25] Luciano Fonseca, Craig Brown, Brian Calder, Larry Mayer, and Yuri Rzhanov. Angular range analysis of acoustic themes from Stanton Banks Ireland: A link between visual interpretation and multibeam echosounder angular signatures. *Applied Acoustics*, 70(10):1298–1304, 2009.

[26] Ariell Friedman, Oscar Pizarro, Stefan B Williams, and Matthew Johnson-Roberson. Multi-scale measures of rugosity, slope and aspect from benthic stereo image reconstructions. *PloS one*, 7(12):e50440, 2012.

[27] Cipriano Galindo, Juan-Antonio Fernández-Madrigal, Javier González, and Alessandro Saffiotti. Robot task planning using semantic maps. *Robotics and Autonomous Systems*, 56(11):955–966, 2008.

[28] Alexander N Gavrilov and Iain M Parnum. Fluctuations of seafloor backscatter data from multibeam sonar systems. *Oceanic Engineering, IEEE Journal of*, 35(2):209–219, 2010.

[29] Yogesh Girdhar, Philippe Giguere, and Gregory Dudek. Autonomous Adaptive Underwater Exploration using Online Topic Modeling. In *Experimental Robotics*, pages 789–802, 2013.

[30] Yogesh Girdhar, David Whitney, and Gregory Dudek. Curiosity based exploration for learning terrain models. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 578–584, 2014.

[31] Yogesh Girdhar, Walter Cho, Matthew Campbell, Jesus Pineda, Elizabeth Clarke, and Hanumant Singh. Anomaly detection in unstructured environments using bayesian nonparametric scene modeling. *arXiv preprint arXiv:1509.07979*, 2015.

[32] Alan Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6645–6649, 2013.

[33] Yuhong Guo and Russell Greiner. Optimistic Active-Learning Using Mutual Information. In *International Joint Conference on Artificial Intelligence*, volume 7, pages 823–829, 2007.

[34] G E Hinton, S Osindero, and Y W Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006.

[35] Geoffrey Hinton. A practical guide to training restricted Boltzmann machines. Technical report, Department of Computer Science, University of Toronto, 2010.

[36] Geoffrey E Hinton. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800, 2002.

[37] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[38] Aapo Hyvärinen. Estimation of non-normalized statistical models by score matching. In *Journal of Machine Learning Research*, pages 695–709, 2005.

[39] D Ierodiaconou, J Monk, A Rattray, L Laurenson, and VL Versace. Comparison of automated classification techniques for predicting benthic biological communities using hydroacoustics and video observations. *Continental Shelf Research*, 31(2):28–38, 2011.

[40] Go Irie, Dong Liu, Zhenguo Li, and Shih-Fu Chang. A Bayesian Approach to Multimodal Visual Dictionary Learning. In *IEEE Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 329–336, 2013.

[41] Ajay J Joshi, Fatih Porikli, and Nikolaos Papanikolopoulos. Multi-class active learning for image classification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2372–2379, 2009.

[42] Jeffrey W Kaeli, John J Leonard, and Hanumant Singh. Visual summaries for low-bandwidth semantic mapping with autonomous underwater vehicles. In *Autonomous Underwater Vehicles (AUV), 2014 IEEE/OES*, pages 1–7. IEEE, 2014.

[43] Hanna Kamyshanska and Roland Memisevic. The Potential Energy of an Autoencoder. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8828(1):720–728, 2014.

[44] AJ Kenny, I Cato, M Desprez, G Fader, RTE Schüttenhelm, and J Side. An overview of seabed-mapping technologies in the context of marine habitat classification. *ICES Journal of Marine Science: Journal du Conseil*, 60(2): 411–418, 2003.

[45] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations*, 2014.

[46] Vladimir E Kostylev, Brian J Todd, Gordon B J Fader, R C Courtney, Gordon D M Cameron, and Richard A Pickrill. Benthic habitat mapping on the Scotian Shelf based on multibeam bathymetry, surficial geology and sea floor photographs. *Marine Ecology Progress Series*, 219:121–137, 2001.

[47] Vladimir E Kostylev, Johan Erlandsson, Mak Yiu Ming, and Gray A Williams. The relative importance of habitat complexity and surface area in assessing biodiversity: fractal application on rocky shores. *Ecological Complexity*, 2(3):272–286, 2005.

[48] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet Classification with Deep Convolutional Neural Networks. *Advances In Neural Information Processing Systems*, pages 1–9, 2012.

[49] Clayton Kunz and Hanumant Singh. Map building fusing acoustic and visual information using autonomous underwater vehicles. *Journal of field robotics*, 30(5):763–783, 2013.

[50] Kevin Lai, Liefeng Bo, and Dieter Fox. Unsupervised feature learning for 3D scene labeling. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3050–3057, 2014.

[51] Hugo Larochelle and Yoshua Bengio. Classification using discriminative restricted Boltzmann machines. *Proceedings of the 25th International Conference on Machine learning*, pages 536–543, 2008.

[52] H Lee, R Grosse, R Ranganath, and A Y Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *Proceedings of the 26th International Conference on Machine learning*, pages 609–616, 2009.

[53] Honglak Lee, C Ekanadham, and Andrew Y Ng. Sparse deep belief net model for visual area V2. *Advances in Neural Information Processing Systems*, 19: 1–8, 2007.

[54] Honglak Lee, Peter Pham, Yan Largman, and Andrew Y Ng. Unsupervised feature learning for audio classification using convolutional deep belief networks. In *Advances in neural information processing systems*, pages 1096–1104, 2009.

[55] Ian Lenz, Honglak Lee, and Ashutosh Saxena. Deep learning for detecting robotic grasps. *The International Journal of Robotics Research*, 34(4-5): 705–724, 2015.

[56] Vanessa Lucieer and Geoffroy Lamarche. Unsupervised fuzzy classification and object-based image analysis of multibeam data to map deep water substrates, Cook Strait, New Zealand. *Continental Shelf Research*, 31(11): 1236–1247, 2011.

[57] Vanessa Lucieer, Nicole A Hill, Neville S Barrett, and Scott Nichol. Do marine substrates 'look' and 'sound' the same? Supervised classification of multibeam acoustic data using autonomous underwater vehicle images. *Estuarine, Coastal and Shelf Science*, 117:94–106, 2013.

[58] David J C Mackay. *Information Theory, Inference, and Learning Algorithms*. Cambridge university press, 2003.

[59] Alexei Makarenko, Stefan B Williams, Frederic Bourgault, Hugh F Durrant-Whyte, and Others. An experiment in integrated exploration. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 1, pages 534–539, 2002.

[60] Junhua Mao, Wei Xu, Yi Yang, Jiang Wang, and AL Yuille. Explain images with multimodal recurrent neural networks. *arXiv preprint arXiv:1410.1090*, pages 1–9, 2014.

[61] M S A Marcos, M Soriano, and C Saloma. Classification of coral reef images from underwater video using neural networks. *Optics Express*, 13(22): 8766–8771, 2005.

[62] Ezequiel M. Marzinelli, Stefan B. Williams, Russell C. Babcock, Neville S. Barrett, Craig R. Johnson, Alan Jordan, Gary A. Kendrick, Oscar R. Pizarro, Dan A. Smale, and Peter D. Steinberg. Large-scale geographic variation in distribution and abundance of australian deep-water kelp forests. *PLoS ONE*, 10:1–21, 02 2015.

[63] Roland Memisevic and Geoffrey E Hinton. Learning to represent spatial transformations with factored higher-order Boltzmann machines. *Neural computation*, 22(6):1473–92, 2010.

[64] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing ATARI with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

[65] Grégoire Montavon and Klaus-Robert Müller. Learning Feature Hierarchies with Centered Deep Boltzmann Machines. *arXiv preprint arXiv:1203.3783*, (1998):1–16, 2012.

[66] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[67] Vinod Nair and Geoffrey E Hinton. Implicit mixtures of Restricted Boltzmann Machines. In *Advances in Neural Information Processing Systems*, pages 1145–1152, 2009.

[68] Vinod Nair and Geoffrey E Hinton. 3D object recognition with deep belief nets. *Advances in Neural Information Processing Systems*, 22:1339–1347, 2009.

[69] Jiquan Ngiam, Andrew Y Ng, Aditya Khosla, Mingyu Kim, Juhan Nam, and Honglak Lee. Multimodal deep learning. In *Proceedings of the 28th Annual Int. Conf. on Machine Learning*, pages 689–696, 2011.

[70] Yagyensh Chandra Pati, Ramin Rezaiifar, and P S Krishnaprasad. Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In *Proceedings of the 27th Asilomar Conference on Signals, Systems and Computers*, pages 40–44, 1993.

[71] Andrzej Pronobis and Patric Jensfelt. Large-scale semantic mapping and reasoning with heterogeneous modalities. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3515–3522, 2012.

[72] Andrzej Pronobis, O Martinez Mozos, Barbara Caputo, and Patric Jensfelt. Multi-modal semantic place classification. *The International Journal of Robotics Research*, 2009.

[73] Marc'Aurelio Ranzato. *Unsupervised Learning of Feature Hierarchies*. PhD thesis, 2009.

[74] Marc'Aurelio Ranzato, Fu Jie Huang, Y-Lan Boureau, and Yann LeCun. Unsupervised Learning of Invariant Feature Hierarchies with Applications to Object Recognition. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2007.

[75] Alex Rattray, Daniel Ierodiaconou, and Tim Womersley. Wave exposure as a predictor of benthic habitat distribution on high energy temperate reefs. *Frontiers in Marine Science*, 2:8, 2015.

[76] Salah Rifai, Pascal Vincent, Xavier Muller, Xavier Glorot, and Yoshua Bengio. Contractive auto-encoders: Explicit invariance during feature extraction. In *Proceedings of the 28th International Conference on Machine Learning*, pages 833–840, 2011.

[77] Paul Rigby, Oscar Pizarro, and Stefan B Williams. Toward adaptive benthic habitat mapping using Gaussian process classification. *Journal of Field Robotics*, 27(6):741–758, 2010.

[78] Yuri Rzhanov, Luciano Fonseca, and Larry Mayer. Construction of seafloor thematic maps from multibeam acoustic backscatter angular response data. *Computers & Geosciences*, 41:181–187, 2012.

[79] Ruslan Salakhutdinov and Iain Murray. On the quantitative analysis of Deep Belief Networks. In *Proceedings of the 25th international conference on Machine learning*, pages 872–879, 2008.

[80] Ruslan Salakhutdinov, Andriy Mnih, and Geoffrey Hinton. Restricted Boltzmann machines for collaborative filtering. In *Proceedings of the 24th International Conference on Machine learning*, volume 227, pages 791–798, New York, New York, USA, 2007.

[81] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, pages 1–8, 2013.

[82] Hanumant Singh, Ali Can, Ryan Eustice, Steve Lerner, Neil McPhee, Oscar Pizarro, and Chris Roman. Seabed AUV offers new platform for high-resolution imaging. *Transactions American Geophysical Union*, 85(31): 289–296, 2004.

[83] Kihyuk Sohn, Wenling Shang, and Honglak Lee. Improved multimodal deep learning with variation of information. *Advances in Neural Information Processing Systems*, pages 2141–2149, 2014.

[84] M Spinoccia. Bathymetry grids of south east Tasmania shelf. 2011.

[85] Nitish Srivastava and Ruslan Salakhutdinov. Multimodal Learning with Deep Boltzmann Machines. *Advances in Neural Information Processing Systems*, 25:1–9, 2012.

[86] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[87] Cyrill Stachniss, Giorgio Grisetti, and Wolfram Burgard. Information Gain-based Exploration Using Rao-Blackwellized Particle Filters. In *Robotics: Science and Systems*, pages 65–72, 2005.

[88] D M Steinberg, S B Williams, O Pizarro, and M V Jakuba. Towards autonomous habitat classification using Gaussian Mixture Models. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4424–4431, 2010.

[89] Daniel Matthew Steinberg. *An Unsupervised Approach to Modelling Visual Data*. PhD thesis, University of Sydney, 2013.

[90] Graham W. Taylor and Geoffrey E. Hinton. Factored conditional restricted Boltzmann Machines for modeling motion style. *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 1–8, 2009.

[91] H B Van Rein, C J Brown, Rory Quinn, and J Breen. A review of sublittoral monitoring methods in temperate waters: a focus on scale. *Underwater Technology*, 28(3):99–113, 2009.

[92] Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural computation*, 23(7):1661–1674, 2011.

[93] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th Int. Conf. on Machine learning*, pages 1096–1103, 2008.

[94] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *The Journal of Machine Learning Research*, 11:3371–3408, 2010.

[95] Stefan Wager, Sida Wang, and Percy S Liang. Dropout training as adaptive regularization. In *Advances in neural information processing systems*, pages 351–359, 2013.

[96] Nan Wang, Jan Melchior, and Laurenz Wiskott. An analysis of Gaussian-binary restricted Boltzmann machines for natural images. In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*, pages 287–292, 2012.

[97] Max Welling, Michal Rosen-Zvi, and Geoffrey E Hinton. Exponential family harmoniums with an application to information retrieval. In *Advances in neural information processing systems*, pages 1481–1488, 2004.

[98] S Williams, O Pizarro, M Jakuba, and N Barrett. AUV benthic habitat mapping in South Eastern Tasmania. In *Field and Service Robotics*, pages 275–284, 2010.

[99] Stefan B Williams, Oscar R Pizarro, Michael V Jakuba, Craig R Johnson, Neville S Barrett, Russell C Babcock, Gary A Kendrick, Peter D Steinberg, Andrew J Heyward, Peter J Doherty, Ian Mahon, Johnson-Roberson Matthew, Steinberg Daniel, and Ariell Friedman. Monitoring of benthic reference sites: Using an autonomous underwater vehicle. *Robotics & Automation Magazine*, 19(1):73–84, 2012.

[100] Margaret F J Wilson, Brian O'Connell, Colin Brown, Janine C Guinan, and Anthony J Grehan. Multiscale terrain analysis of multibeam bathymetry data for habitat mapping on the continental slope. *Marine Geodesy*, 30(1-2):3–35, 2007.

[101] Markus Wulfmeier, Peter Ondruska, and Ingmar Posner. Deep Inverse Reinforcement Learning. *arXiv preprint arXiv:1507.04888*, 2015.

[102] Jianchao Yang, Kai Yu, Yihong Gong, and Thomas Huang. Linear Spatial Pyramid Matching using Sparse Coding for Image Classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1794–1801, 2009.

[103] Dana R Yoerger, Albert M Bradley, Barrie B Walden, Hanumant Singh, and Ralf Bachmayer. Surveying a subsea lava flow using the Autonomous Benthic Explorer (ABE). *International Journal of Systems Science*, 29(10):1031–1044, 1998.

[104] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *Computer Vision–ECCV 2014*, pages 818–833. Springer, 2014.

[105] Guanyu Zhou, Kihyuk Sohn, and Honglak Lee. Online Incremental Feature Learning with Denoising Autoencoders. In *International Conference on Artificial Intelligence and Statistics*, pages 1453–1461, 2012.