Generative Probabilistic Models for Object Segmentation

S. M. Ali Eslami

Doctor of Philosophy

Institute for Adaptive and Neural Computation
School of Informatics
University of Edinburgh

2013

Paleolithic painting of a horse at Lascaux, estimated to be 17,300 years old.
Photo courtesy of Aujoulat (2005).

Abstract

One of the long-standing open problems in machine vision has been the task of 'object segmentation', in which an image is partitioned into two sets of pixels: those that belong to the object of interest, and those that do not. A closely related task is that of 'parts-based object segmentation', where additionally each of the object's pixels are labelled as belonging to one of several predetermined parts.

There is broad agreement that segmentation is coupled to the task of object recognition. Knowledge of the object's class can lead to more accurate segmentations, and in turn accurate segmentations can be used to obtain higher recognition rates. In this thesis we focus on one side of this relationship: given the object's class and its bounding box, how accurately can we segment it?

Segmentation is challenging primarily due to the huge amount of variability one sees in images of natural scenes. A large number of factors combine in complex ways to generate the pixel intensities that make up any given image. In this work we approach the problem by developing generative probabilistic models of the objects in question. Not only does this allow us to express notions of variability and uncertainty in a principled way, but also to separate the problems of model design and inference.

The thesis makes the following contributions: First, we demonstrate an explicit probabilistic model of images of objects based on a latent Gaussian model of shape. This can be learned from images in an unsupervised fashion. Through experiments on a variety of datasets we demonstrate the advantages of explicitly modelling shape variability.

We then focus on the task of constructing more accurate models of shape. We present a type of layered probabilistic model that we call a *Shape Boltzmann Machine* (SBM) for the task of modelling foreground/background (binary) and parts-based (categorical) shapes. We demonstrate that it constitutes the state-of-the-art and characterises a 'strong' model of shape, in that samples from the model look realistic and that it generalises to generate samples that differ from training examples.

Finally, we demonstrate how the SBM can be used in conjunction with an appearance model to form a fully generative model of images of objects. We show how parts-based object segmentations can be obtained simply by performing probabilistic inference in this joint model. We apply the model to several challenging datasets and find that its performance is comparable to the state-of-the-art.

Lay summary

One of the long-standing open problems in artificial intelligence has been the task of parts-based object segmentation, in which an image of an object is partitioned into different sets of pixels, each corresponding to either one of several predetermined object parts, or to the image background. This task is thought to be important for recognition, but also for interaction: a robot will need to know the precise location of an object and its parts to be able to interact with it.

Segmentation is challenging primarily due to the huge amount of variability one sees in images of natural scenes. A large number of factors combine in complex ways to generate the pixel intensities that make up any given image. These factors include, but are not limited to, object pose, appearance and shape, camera pose and scene illumination.

When the objects colours are near constant in the dataset, e.g. in videos, statistics of their pixel colours have been used to guide segmentation. However, for many datasets of interest object appearances are too variable to be modelled with accuracy. In this thesis we consider probabilistic models that allow us to incorporate knowledge about shapes for the segmentation task.

First, we present a framework in which separate models of shape and appearance can be reasoned about simultaneously. This allows us to learn probabilistic models of the two directly from training images, and to combine them to obtain accurate segmentations of unseen images.

Second, we focus on developing a model of shapes that we call the *Shape Boltzmann Machine* (SBM). We demonstrate how, using the SBM, accurate shape models can be learned from very small training datasets. Through qualitative and quantitative experiments we show that the SBM is a strong model of shape, in that samples from the model look realistic and it generalises to generate samples that differ from training examples.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*S. M. Ali Eslami*)

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

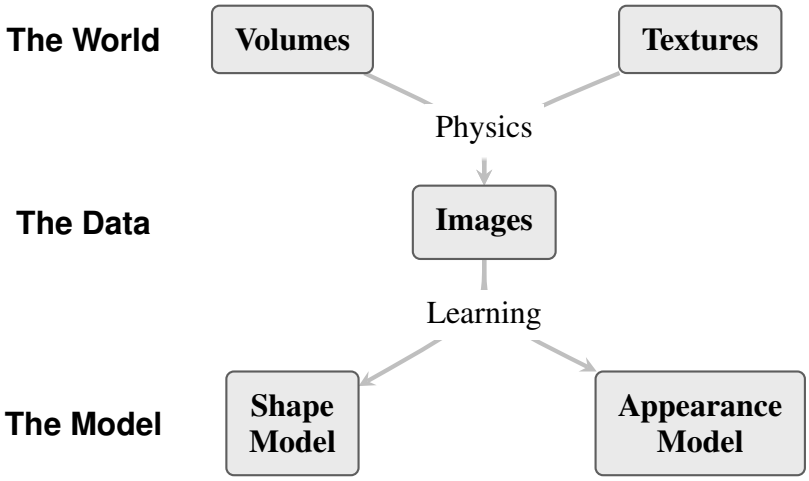


Figure 1.1: Modelling the world through shapes and appearances.

As a scientific discipline, computer vision is concerned with the theory behind artificial systems that extract useful information from images. For example in a classification task, the presence or absence of an object class is determined in an image. In a localisation task, an object's 2D position or its 3D pose is found in the scene. In an identification task, a *specific* instance of an object class is recognised in an image.

Research on computer vision tasks dates back to the earliest days of computing (see e.g. Crevier, 1993 for a review). Despite its history, and despite the huge growth of interest it has seen in recent years, humans still consistently outperform state-of-the-art computer vision algorithms at most tasks, both in terms of accuracy and speed.

The difficulty of computer vision tasks lies primarily in the huge amount of variability

one sees in images of natural scenes. A large number of factors combine, often in complex ways, to generate the pixel intensities that make up any given image. These factors include, but are not limited to, object pose, appearance and shape, camera pose and scene illumination.

As an example, consider a scene consisting of a cat and a ball of yarn. Depending on the cat's position relative to the ball and the camera, hugely different images can be taken of the scene. One can imagine the cat or the yarn (or both) being partially or completely missing from the image if they are occluded or outside the camera's field of view. Additionally, different object colours (whether the cat is yellow or black), poses (whether it is sitting or standing) and subclasses (whether it is a Persian or Russian breed) can lead to large perturbations of the image's pixel intensities.

It has been known for some time that the so-called 'inverse problem', that of extracting information about the real world from the patterns of light that fall onto a camera's sensor, is ill-posed (e.g. Horn, 1977; Poggio et al., 1985). It is generally agreed that in order to obtain accurate inferences, there is a need for internal *models* that combine sensory evidence with *prior knowledge* about properties of the world.

Most existing models of visual scenes operate only at the lowest-level: that of the pixel. Such models focus on capturing local statistical regularities of image pixels and often lack the capabilities required to accurately explain all of the variability seen in natural images. At the other end of the spectrum, it may one day be possible to recognise images in terms of high-level collections of 3D objects. We know how to generate from these models (this is what graphics engines do), and we can imagine how accurate inference of such a model's variables for a given image would lead to an accurate understanding of that image. However, inference at such a scale is still considered to be computationally intractable.

An opportunity exists to explore models in the space between these two kinds of approaches. Specifically, to design models that understand visual scenes as collections of semantically meaningful objects, whilst, for computational reasons, aiming to keep the object models themselves as simple as possible.

In order to reason about images as collections of objects, the computer will have to be able to reliably classify, locate and segment the objects in the scene. All three tasks remain challenging, partly due to their inherently interrelated nature. We note that for a computer system to understand visual scenes well, it should at least be able to perform

each one of these tasks when given the ground truth values of the other two. In this thesis we focus on the task of object segmentation *given* the object's class and extent in the image.

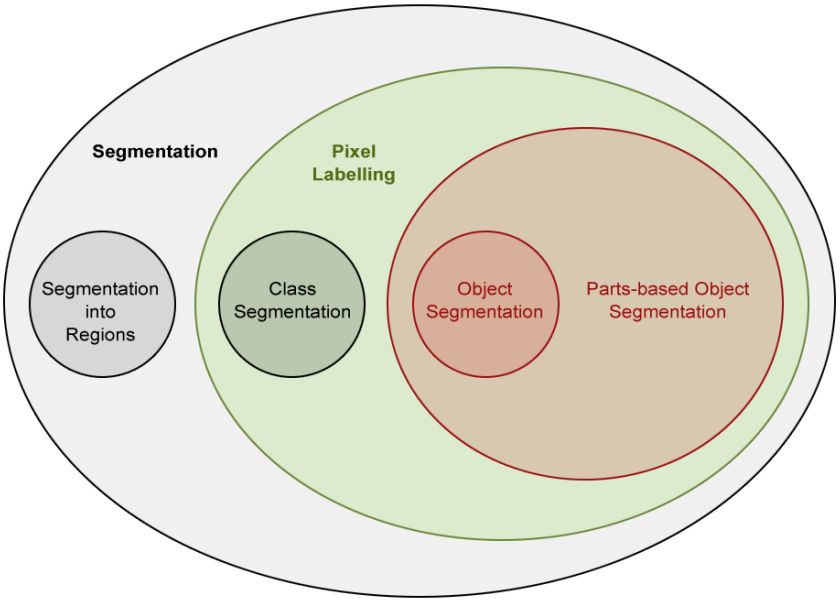
The term 'segmentation' has been used to refer to at least two similar but distinct tasks in computer vision literature. The first, is to partition an image into an *arbitrary* number of visually coherent regions. The second, which is also sometimes referred to as 'pixel labelling', is to classify each pixel in an image into a *fixed* number of categories. For instance, the labels may correspond to different object classes (as in the PASCAL Visual Object Challenge, Everingham et al., 2010). In this thesis we focus on the following types of pixel labelling:

- **Object segmentation:** To assign each pixel to one of only two labels, one corresponding to a foreground object of interest and the other to the background, or
- **Parts-based object segmentation:** To assign each pixel to one of $K + 1$ labels, K corresponding to the different parts that make up the foreground object and one corresponding to the background.

See Fig. 1.2 for an illustration of these different tasks.

There is a rich history of work on segmentation by only considering low-level pixel statistics of segmentations (in the form of Markov random fields and conditional random fields, see e.g. Boykov and Jolly, 2001; Rother et al., 2004). However, on their own, these statistics often fail to provide the amount of information needed to obtain accurate segmentations. To see why, one only has to examine the kinds of images that such approaches find difficult to segment. Errors can typically be attributed to a lack of high-level, cross-image understanding about the objects in question.

When the objects' colours are near constant in the dataset (e.g. in videos), statistics of their *appearances* have been used as the primary drivers of segmentation algorithms (e.g. Cootes et al., 2001; Frey et al., 2003; Williams and Titsias, 2004; Cemgil et al., 2005). However, for many datasets of interest object colours are too variable to be modelled with sufficient accuracy. Whilst the aforementioned methods typically reason about shape in one way or another, their models of shape are often not precise enough to drive segmentation when applied to challenging images.



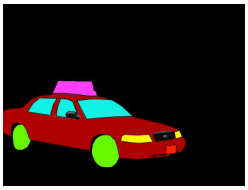
(a)

(b)

(c)

(d)

(e)



(f)

Figure 1.2: **Comparison of the segmentation tasks**. (a) Venn diagram of the different segmentation tasks. The ones considered in this thesis have been highlighted in red. (b) A typical visual scene. Photo credits: http://panoramio.com/photo/4311184. (c) A partitioning of the image into visually coherent regions. (d) The assignment of each pixel in the image to three groups, each corresponding to a different class of object in the scene (class segmentation). (e) The assignment of each pixel to either the foreground or the background (object segmentation), or (f) to either one part of the foreground object or to the background (parts-based object segmentation).

(a) (b)

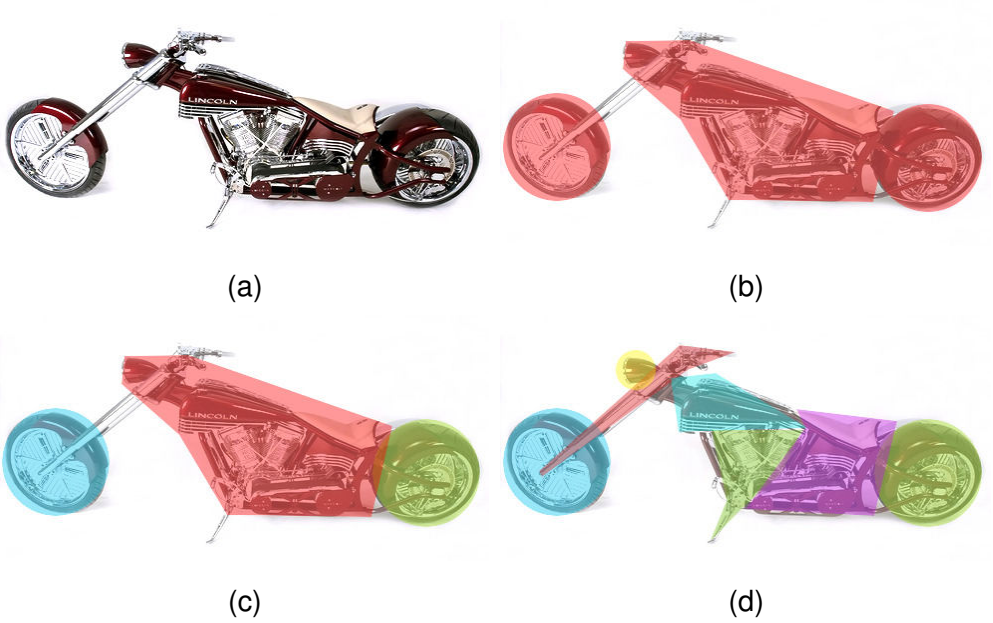
Figure 1.3: **Two different kinds of shape models**. (a) A three dimensional human manikin. (b) A two dimensional human template. Photo credits: Jack Richeson Human Manikins and Template Designs TD1735A.

One of the central goals of this thesis will be to design models that incorporate more accurate prior knowledge about shapes for the segmentation task. By carefully exploiting models of shapes, we hope to be able to apply our models to datasets of increasing complexity.

1.1 Shapes, appearances and parts

We live in a three dimensional world with three dimensional objects. Ideally, we would also like to reason about each object using a three dimensional model, where object shapes are represented as volumes, meshes or point clouds (e.g. Fig. 1.3a), and their appearances are represented as texture maps. Whilst, in theory, we know how to do inference in three dimensional models of the world, it rapidly becomes infeasible for all but the most simple and constrained of such models (see e.g. Cashman and Fitzgibbon, 2012). For this reason we consider an approximation to the real world in which object shapes are reasoned about in a two dimensional space for each viewpoint (e.g. as in Fig. 1.3b), and we model different viewpoints separately (e.g. with the use of a mixture).

In this view, an object's appearance is the collection of pixels in image space that lie within the object's boundaries. For many objects of interest, appearance variability is best explained by considering the *parts* that make up the object. For example, a pedestrian's appearance is most succinctly defined by first describing the precise location and extent of her hair, skin, shirt, trousers and shoes, and second, describing each of



(a) (b)

(c) (d)

Figure 1.4: **What is the right level of granularity for parts?**. (a) An image of a motorcycle. Photo credits: Orange County Choppers. (b) The motorcycle, represented by a single 'part'. (c) The motorcycle represented by three parts: two wheels and a body. (d) The motorcycle represented by many parts: two wheels, a headlight, a saddle, an engine and the front forks and handlebars.

their colours, textures and patterns.

There are a number of issues one needs to be aware of when considering such an approach. The first is that not all object classes can be easily decomposed into parts. Consider as an example the loaf-of-bread object class discussed by Ullman (1996): Although we can semantically separate it into multiple slices, doing so will typically not be of any use for us in describing its appearance. In such cases, we would simply want to assume the object to be composed of a single part. The second issue is that there may potentially be a huge number of parts that combine to explain the precise appearance of a specific instance of a class (e.g. a motorcycle is composed of thousands of exposed semantic parts). Although it is possible to reason about the segmentation in this way, our aim will be to reason about parts at a coarse enough level for it to be meaningful to have *prior* knowledge about the parts and their shapes before any test data has even been seen – in this case, for example, the knowledge that motorcycles are typically composed of two wheels, a chassis and a body (see Fig. 1.4).

Related to the issue of part granularity is that of part sharing. Some approaches to

(a) (b) (c)

Figure 1.5: **The case against shared parts**. (a) An image of a horseshoe. (b) The horseshoe represented by a single horseshoe-looking part. (c) The horseshoe represented by generic rectangular parts. When forced to explain an object using generic parts, the parts may lose their utility in explaining the object's appearance.

parts-based modelling encourage the sharing of parts between classes of objects. By doing so, these techniques arrive at a smaller number of 'primitives' – generic parts that can be seen in many classes of objects (see e.g. Biederman, 1987). However, when forced to share between classes, such models typically opt to learn that parts are very small (see Fig. 1.5 and Torralba et al., 2007), thereby losing their utility in describing the objects' appearances.

Finally, we note that it will also be beneficial to allow for a certain amount of flexibility in each part's shape. To see why, consider the dataset of images of hammers in Fig. 1.6. A model that represents the hammers with two parts will have to be able to account for the variability of the heads and shafts in the dataset.

To summarise, for both objects and parts, we consider 'shape' to be some description of their extents in image space, and 'appearance' to be some description of the pixel values within their boundaries. Shapes and appearances vary considerably in most natural image datasets of interest, therefore it will be necessary to build models that can accurately account for, and reason about, both kinds of variability.

Figure 1.6: **The case for flexible shape models**. Consider a model in which hammers are represented by two parts: a head and a shaft. In order to remain faithful to the data, such a model will have to account for the variability of head shapes.

1.2 Probabilistic generative models

In order to reason about higher-level image concepts such as objects, parts, shapes and appearances, it is now becoming increasingly commonplace to frame vision problems as that of inference in probabilistic models (see e.g. Forsyth and Ponce, 2011 for a review). Not only does this allow us to express notions of variability and uncertainty in a principled way, but also to separate the problems of model design and inference.

Generative probabilistic models additionally allow us to sample from our trained models. By visually inspecting the quality of these samples, we begin to understand exactly what the model has learned, and we identify weak-points in our models that can guide our future research efforts. This approach is particularly well-suited to the visual domain due to the speed and accuracy with which our brains can process and evaluate visual information.

Generative models are also flexible with regards to training data, in that they are amenable to unsupervised and semi-supervised learning. Again this is a particularly useful trait in the visual domain where labelled training data is expensive and rare.

Within this setting, the ideal would be to construct models that are expressive enough to generate high-quality samples of images of objects (i.e. ones that look realistic but also generalise in the shapes, appearances and locations of objects within the image), but also simple enough for inference and learning to remain tractable. Much of this thesis is concerned with finding a suitable ‘middle-ground’ in this space.

One important premise of this thesis is that it is beneficial, and indeed more natural, to model object shapes and appearances independently. By building separate models of shape and appearance, we hope to be able to break the problem of building a generative probabilistic model of images into smaller, more manageable pieces. Note that, if required, it will still be possible to model correlations between shape and appearance by introducing additional random variables at the top-most levels of the model (see e.g. Active Appearances Maps of Cootes et al., 2001).

1.3 Outline of the thesis

In this thesis we will formally define what is meant by shapes and appearances, show how models of the two can be learned from data, and describe how they can be combined to extract information from images (e.g. for segmentation). The remainder of this document is structured as follows:

Chapter 2 presents an overview and comparison of existing work in the area of object segmentation and parts-based modelling.

Chapter 3 demonstrates an explicit probabilistic model of images of objects based on a latent Gaussian model of shape. We present a novel parts-based image representation that learns from unlabelled images that exhibit variability in both the shapes and appearances of objects. Through experiments on a variety of datasets we demonstrate the advantages of explicitly modelling shape variability. We also show that the model's latent representations can be interpreted as 'parsings' of images, and that these parsings are accurate enough to be used even for tasks like fine-grained categorisation. Finally, we apply the model to the object segmentation task, and find that its performance is comparable to that of the state-of-the-art on a number of benchmark datasets. The work presented in this chapter has been published as follows:

- Eslami, S. M. A. and Williams, C. K. I. (2011). Factored Shapes and Appearances for Parts-based Object Understanding. In *British Machine Vision Conference (BMVC)*

Chapter 4 focusses on the task of constructing accurate models of shapes. We present a type of Deep Boltzmann Machine (Salakhutdinov and Hinton, 2009) that we call a *Shape Boltzmann Machine* (SBM) for the task of modelling foreground/background (binary) shapes. We show that the SBM characterises a 'strong' model of shape, in that

samples from the model look realistic and that it generalises to generate samples that differ from training examples. Finally, we demonstrate that the SBM learns distributions that are qualitatively and quantitatively better than existing models at this task. The work presented in this chapter has been published as follows:

- Eslami, S. M. A., Heess, N., and Winn, J. (2012). The Shape Boltzmann Machine: a Strong Model of Object Shape. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*
- Eslami, S. M. A., Heess, N., Williams, C. K. I., and Winn, J. (2013). The Shape Boltzmann Machine: a Strong Model of Object Shape. *International Journal of Computer Vision (IJCV) (under review)*

Chapter 5 demonstrates how the SBM can be used in conjunction with an appearance model to form a fully generative model of images of objects. We show how parts-based object segmentations can be obtained simply by performing probabilistic inference in this joint model. We apply the model to several challenging datasets and find that its performance is comparable to the state-of-the-art. The work presented in this chapter has been published as follows:

- Eslami, S. M. A. and Williams, C. K. I. (2012). A Generative Model for Parts-based Object Segmentation. In *Advances in Neural Information Processing Systems 25*

Chapter 6 summarises the results presented in the thesis, outlines directions for future work and concludes with a discussion.

Chapter 2

Background

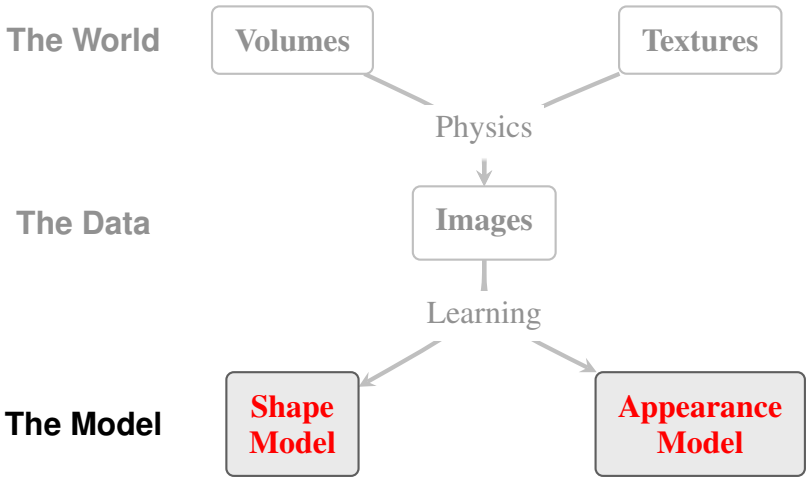


Figure 2.1: A survey of existing models of shapes and appearances.

Segmentation can be cast as an instance of the canonical classification problem in statistical pattern recognition. The probabilistic approach is well suited to this task, as it provides a principled framework with which models can be built that 'learn' to cope with the task's complexities.

Given *the image* $\mathbf{X}$ consisting of D pixels $\mathbf{x}_d$ ($d = 1...D$) in some feature space (e.g. RGB pixel values or SIFT descriptor values, Lowe, 2004), we wish to obtain a label $\mathbf{s}_d$ for each pixel. The assignment of pixel d to each of the possible groups (one for the object or each of its parts and one for the background) is determined by $\mathbf{s}_d$'s value. We will refer to the collection of all labellings in matrix form $\mathbf{S}$ as *the segmentation* (see Fig. 2.2).

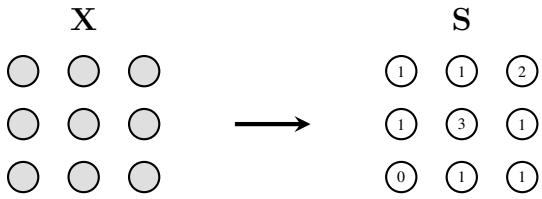


Figure 2.2: **The canonical segmentation task**. Given an observed image $\mathbf{X}$ consisting of D pixels, the goal is to obtain a good segmentation $\mathbf{S}$ – the assignment of each pixel to a segmentation group. Here we illustrate a possible setting of $\mathbf{S}$, one which assigns each pixel to four different groups.

This classification problem can be tackled in a number of different ways. Approaches that explicitly or implicitly model the joint probabilistic distribution of images and segmentations $p(\mathbf{X}, \mathbf{S})$ are known as *generative* models, since by sampling from them it is possible to generate synthetic images. Alternatively, one can directly model the posterior probability of assignments $p(\mathbf{S}|\mathbf{X})$ directly with so-called *discriminative* models (see Bishop, 2006 for further details).

A probabilistic segmentation *technique* is a suite of algorithms for inference and learning in such probabilistic models. A segmentation technique's efficacy is determined by its accuracy and the computational complexity of its learning and inference algorithms.

In this chapter we will review several classes of techniques that have been used for the task of segmentation in the literature. Our intention here is to provide a high-level overview – we will consider more detailed comparisons of methods that are related to the models presented in this thesis in the chapters that follow.

We begin by considering simple unary approaches in Sec. 2.1. We then review techniques that employ *local* random field models to clean up segmentations in Sec. 2.2. We discuss more sophisticated methods that employ *global* shape models in Sec. 2.3, touch on the issue of occlusion in Sec. 2.4 and conclude with a summary in Sec. 2.5.

2.1 Unary methods

Perhaps the most naïve approach to segmentation is to independently assign a label to each pixel in the image using only information derived from its local features (e.g. RGB pixel values or SIFT descriptors, Lowe, 2004). Whilst such approaches have been shown to be effective at segmenting images in which the object's appearance

is roughly constant and visually dissimilar from the background, they often lead to poor results on more challenging datasets where object pixel values can be almost arbitrary. This variability can be due to factors such as scene illumination, appearance heterogeneity and sensor noise. Most of the techniques considered in this survey utilise more sophisticated models to overcome these issues.

2.2 Local, continuity-based methods

Markov random field (MRF) and conditional random field (CRF) models are two families of probabilistic graphical models that have been widely used for segmentation. MRFs were originally introduced as a Bayesian model of grayscale image pixels and were used for image restoration (Geman and Geman, 1984). CRFs were rapidly employed in the context of computer vision (see Kumar and Hebert, 2003 for an early example) after having originally been used to label sequenced data (Lafferty et al., 2001). The main idea behind these techniques is that segmentations with desirable high-level properties can be obtained using mainly low-level constraints on the pixel labellings.

In such methods, a probabilistic graphical model is constructed on the image by considering each pixel and each label as a random variable and connecting sets of label variables with edges. The criterion, or 'goodness of a segmentation', is defined via the *energy* of this graph, and learning and inference algorithms are devised to minimise this energy. The energy function typically consists of *unary potentials* that 'prefer' pixel labellings that are consistent with their corresponding pixels, and *pairwise potentials* that prefer labellings consistent with pairwise statistics of the two pixels in consideration (see Fig. 2.3). In the MRF, the joint distribution of labellings and pixels $p(\mathbf{X}, \mathbf{S})$ is defined to be proportional to $\exp\{-E(\mathbf{X}, \mathbf{S})\}$, where $E(\mathbf{X}, \mathbf{S})$ is called the energy of a joint configuration of the image and segmentation random variables, and the exponential representation is called the *Boltzmann distribution*. By contrast, CRFs model the discriminative distribution on labellings $p(\mathbf{S}|\mathbf{X})$ directly using the energy function. For a more detailed discussion of MRFs and CRFs see Sec. 4.1.

Many random field segmentation techniques differ only in their specific chosen forms of the unary and pairwise (and possibly higher-order) potentials. For example in Interactive Graph Cuts (IGC) of Boykov and Jolly (2001), human users provide hard

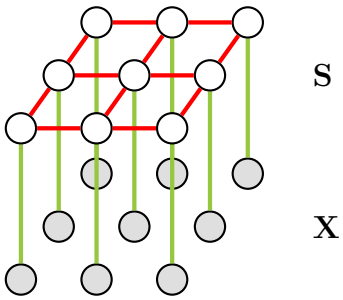
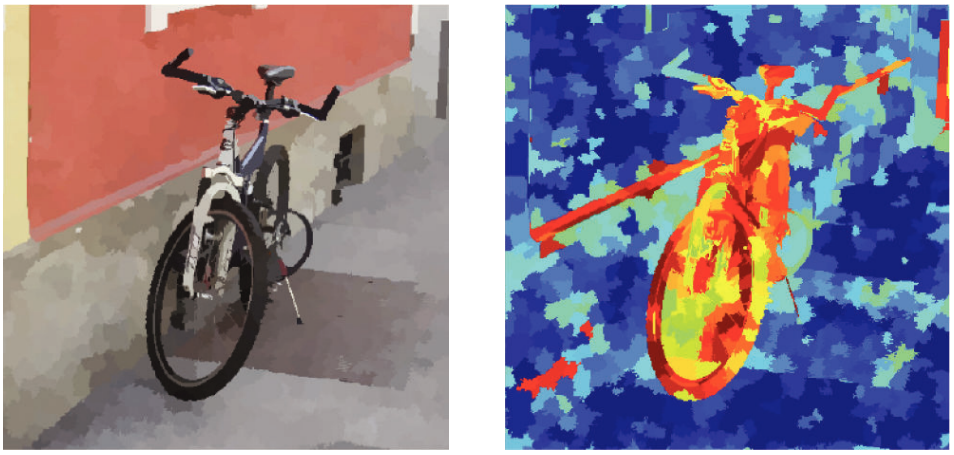


Figure 2.3: **The random field family of models**. The distribution $p(\mathbf{X}, \mathbf{S})$ involves unary potentials between observed pixels and labels (green), and pairwise potentials between pairs of labels (red).

constraints for segmentation by marking certain pixels as 'object' or 'background'. The energy function incorporates this information, along with standard boundary and region cues (in the form of unary and pairwise potentials), to obtain good segmentations. In GrabCut (Rother et al., 2004), the monochrome image model of IGC is replaced for colour by a Gaussian Mixture Model (GMM), and an iterative algorithm is presented for finding the minimum of the energy function. Additionally, Lempitsky et al. (2009) show how a user-provided bounding box on the object can be used as a prior, guiding the algorithm to make segmentations that remain suitably close to the provided bounding box.

More intricate image features have been used in the literature to obtain accurate segmentations within the random field framework. For example in TextonBoost (Shotton et al., 2006), the unary potentials in the CRF include information from a texton map and a location map in addition to the image itself. The texton map – a mapping of each pixel to a cluster of textons – is constructed directly from the image and is designed to efficiently capture local appearance and shape statistics. Using these techniques the model jointly incorporates information about appearance, shape and context.

Some argue that operating at the pixel level is redundant (due to the similarity between neighbouring pixels), and that it can be sensitive to, and limited by, the resolution of the image. A number of techniques have been proposed which overcome these problems by working at a level higher than that of pixels. In He et al. (2006) a CRF is constructed on the result of the partitioning of the image into small but coherent regions called 'superpixels'. The unary potentials in the energy function now capture the statistics of the histograms of features found in these superpixels. Fulkerson et al. (2009) present a similar model with the Superpixel CRF (SPCRF), in which they aggregate the features



(a) (b)

Figure 2.4: **Site labelling with the Superpixel CRF (Fulkerson et al., 2009)**. (a) Instead of operating at the pixel level, superpixels are used as the basic unit of segmentation. (b) Statistical models classify each superpixel as either foreground or background. Here, red indicates high confidence in the superpixel being part of the foreground.

of each superpixel's neighbourhood to obtain good segmentation results on challenging datasets (see Fig. 2.4 for an illustration).

Such random field methods for segmentation have been shown to be extremely effective when provided at least a minimal amount of user-assistance (see Fig. 2.5 for an example). Despite this, random field techniques often fail to provide good segmentations when provided with no human guidance, as their models typically capture low-level, pairwise statistics between pixels and have no knowledge of high-level object shape.

2.3 Global, shape-based methods

Intuitively, *prior* knowledge about an object class' general shape should also provide cues for its segmentation. Consider as an example the 'cow' object class. Since cows have legs, a good segmentation algorithm should take a cow's legs' common spatial positions relative to its body into consideration when segmenting an image of it, even though they may be difficult to distinguish from the background at first glance. There is evidence to suggest that humans use the same kind of knowledge when interpreting visual scenes (Peterson and Gibson, 1993).

(a) (b)

Figure 2.5: **An example of the segmentations produced by GrabCut (Rother et al., 2004)**. Given a loose bounding for the object in consideration (a), accurate segmentations are found (b).

Regularities can often be found in the range of shapes that an instance of an object class takes. However, it is rarely the case that the shapes remain precisely constant throughout a dataset. Shape models typically define explicit or implicit *distributions* on the range of possible shapes their instances take to capture the *variability* of an object class' shape. In order to be useful, shape models need be as specific as possible in defining legal shapes, whilst remaining flexible enough to cope with the class' variability.

One possible way of categorising existing approaches to the problem of modelling shape is to separate them into two groups: methods that predominantly model the object's external outline, and methods that predominantly model its internal structure (the outlines of the parts that constitute the object). In Secs. 2.3.1.1, 2.3.1.2 and 2.3.1.3 we give examples of models that exploit latent variables, parts and fragments respectively to capture the variability of the shape outline. In Sec. 2.3.2.1 we give examples of models that focus on the statistics of the appearances of parts to identify the object's internal structure. In the next chapter, in Sec. 3.3, we provide a detailed discussion of models that attempt to simultaneously model object outline and internal structure using a shared set of parts.

2.3.1 Modelling the object's outline

2.3.1.1 Latent variable models

'Deformable templates' can be seen as a class of statistical models of object shape. These models are defined by exemplars of the object's shape that are learned during training and deform to fit objects in new images. The Active Shape Model (ASM) of Cootes et al. (1995) represents the exemplar *sparsely* as a constellation of points. Using the points' covariance statistics, the shapes' principal modes of variation are determined, and this allows the object's deformations to be described by a small set of linearly independent parameters. An iterative algorithm is used to update the parameters to fit the image (see Fig. 2.6).

In the LOCUS model (Winn and Jojic, 2005), the exemplar is represented *densely* by mask and edge probability maps. A deformation vector field is defined on the two which shifts small patches of the maps to fit objects in new images (see Fig. 2.7). The deformed mask and edge probability maps are then used to guide a local segmentation algorithm similar to that of GrabCut, effectively replacing the human's input.

Similarly in Layered Subspace Models (LSM, Frey et al., 2003) and Stel Components Analysis (SCA, Jojic et al., 2009) object shape is defined by dense probability masks. However, instead of *deforming* the mask to explain the class' shape variability, its variability is explicitly captured using some top down generative procedure (see, e.g., Fig. 2.8). In such latent subspace models the variability of observed variables (in this case, the mask) is represented in terms of a potentially lower number of unobserved variables. The values these factors take can be interpreted as coordinates in some lower-dimensional feature space.

2.3.1.2 Parts-based models

For structured, articulated or highly deformable object classes, it may be more natural (and more efficient) to reason about shape variability in terms of the class' constituent parts (Biederman, 1987). The main idea behind parts-based approaches for shapes is that they combine factorially to generate the object's shape.

There is an extensive history of work on parts-based models in computer vision, dating back almost 40 years. These works differ in a number of ways. First, they differ in the

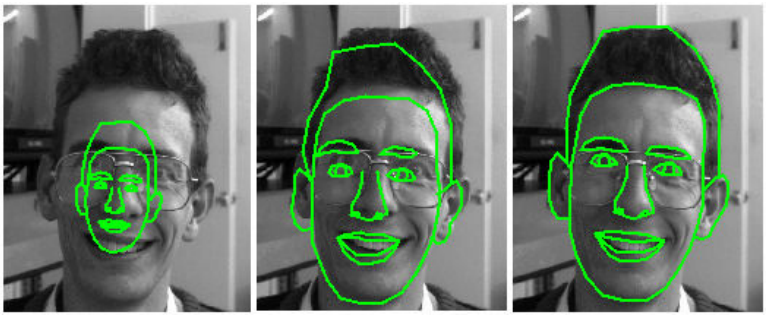


Figure 2.6: **A face exemplar represented as a constellation of points in the Active Shape Model (Cootes et al., 1995)**. The exemplar is iteratively deformed to fit a previously unseen image of a face.

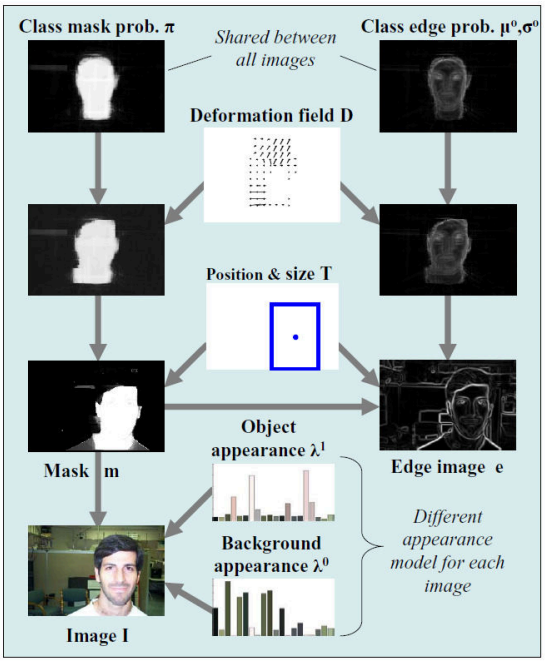


Figure 2.7: **The LOCUS generative model (Winn and Jojic, 2005)**. The class and edge masks are deformed by $\mathbf{D}$ and transformed by $\mathbf{T}$ to match the image.

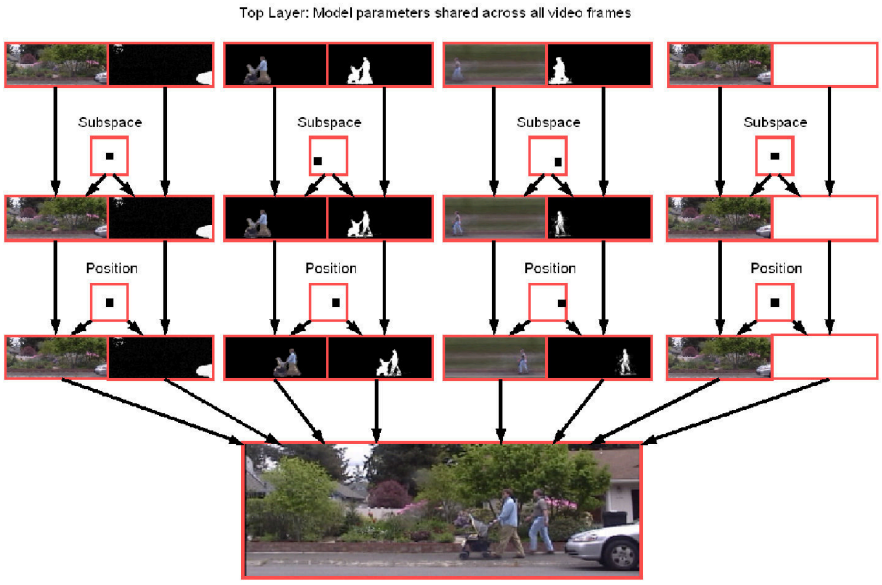


Figure 2.8: **An example of the Layered Subspace Models employed by Frey et al. (2003) to model video frames**. Multiple subspace models (columns) combine to describe the image.

way they define what a 'part' is. Second, they differ in their choice of representation for the parts (for example if they are defined by their shapes or their appearances). For example, Biederman (1987) represents objects a collection of 'geons' – simple 3-dimensional forms such as spheres and cubes – positioned in space, whereas Kannan et al. (2006) represent objects as 'jigsaws' of appearances. Finally, they differ in the way the parts are learnt from data. Whilst most approaches learn the parts for each class of objects separately, others do not (see e.g. Torralba et al., 2007).

One of the earliest examples of parts-based works is the *Pictorial Structure* (PS) of Fischler and Elschlager (1973) – an image model consisting of a collection of parts arranged in a deformable configuration represented by spring-like connections (see Fig. 2.9). More recently, probabilistic interpretations have been provided for such models (Felzenszwalb and Huttenlocher, 2000), and they have been used to great effect for the object classification and detection tasks (Fergus et al., 2003; Sudderth et al., 2008; Felzenszwalb et al., 2009; Zhu et al., 2010).

OBJCUT (Kumar et al., 2005) is an example of a segmentation technique that incorporates such a parts-based model. A structure similar to that of the PS is used as a prior for an MRF constructed on the image's pixels. The MRF's energy function contains unary potentials that incorporate features derived from the shape model. The shape

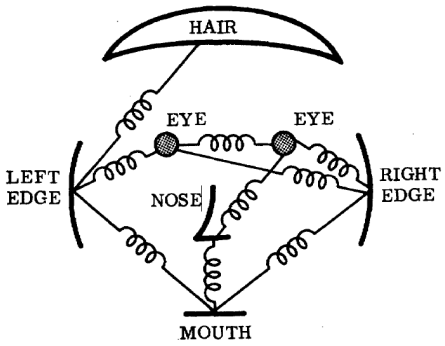


Figure 2.9: **Schematic representation of a Pictorial Structure (Fischler and Elschlager, 1973)**. The face PS indicates the various components and their linkages.

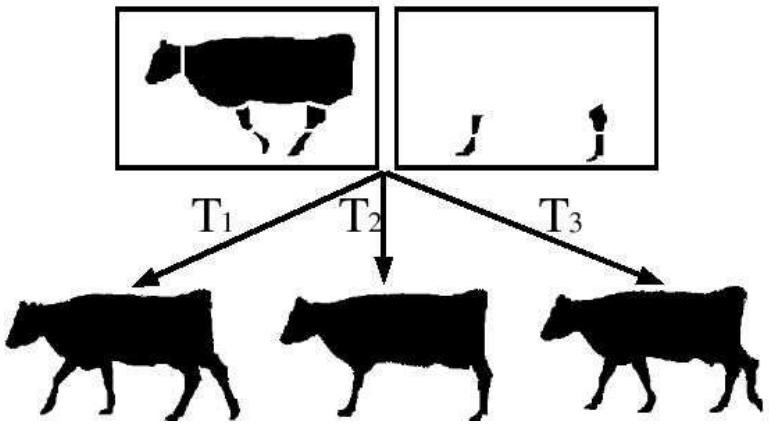


Figure 2.10: **Layered Pictorial Structure of a cow (Kumar et al., 2005)**. The parts combine in layers (top) to generate valid configurations (bottom). These configurations are used as biases in the segmentation MRF.

model, the Layered Pictorial Structure (LPS), is an extension of the PS that represents the object as a composition of outline and texture features in multiple occluding layers. Model parameters are learned by dividing *videos* of an instance of the object class into rigidly moving components and extracting outline and texture information from the components (see Kumar et al., 2004 for details). Given a test image, the best LPS pose for the scene is found and used to guide the MRF's energy minimisation. It is important to note that although the PS uses the spring-like connections between reference parts effectively to explain different instantiations of the same object class, it does not model the variability of the parts themselves.

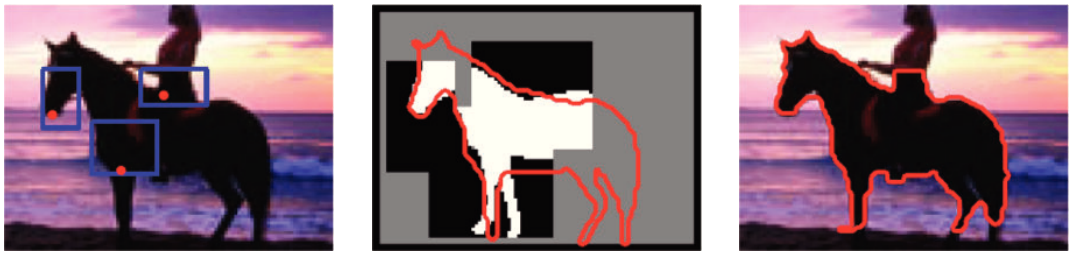
Finally, we mention the Dirichlet Process Mixture CRF (DPMCRF) (Larlus et al., 2009), in which a DPM is used to provide object-class specific biases for a standard

random field model. The DPM models images as a composition of regions or 'blobs', each representing a single object instance. In this way the authors provide local spatial regularisation (using the random field), and at the same time capture more high-level structures (using the Dirichlet process mixture). However, the DPM imposes a very broad prior on the object's presence in any part of the image and the shape of this prior is not in any way dependent on the object's class.

2.3.1.3 Fragments-based models

An example of an alternative approach to parts-based modelling is that of Borenstein et al. (2004). In this work, top-down cues of object shape are generated by using a collection of fragments stored in memory to detect and cover the object. The ground truth labelling of the covering fragments is then used to label the image pixels as foreground or background. The top-down segmentation is combined with bottom-up segmentations using a *cost function* that evaluates compromise segmentations between the two. The cost function penalises segmentations 'far away' from the top-down segmentation, as well as segmentations that separate homogeneous image regions into foreground and background parts. Good segmentations are found by minimising this cost function. Levin and Weiss (2009) formulate a similar approach to the problem within a unified CRF framework (the Fragment CRF or FCRF), which allows them to take into account both bottom-up and top-down cues simultaneously during training. Whereas pure top-down algorithms often require hundreds of fragments, this simultaneous learning procedure yields segmentation algorithms that operate using fewer fragments. At run-time, their algorithm is identical to that of Borenstein et al. (2004) (see Fig. 2.11).

Similarly, an Implicit Shape Model (ISM, Leibe et al., 2004) for an object class consists of a *codebook* of local appearances that are prototypical for the object, and of a spatial probability distribution which specifies where each codebook entry may be found on the object. Small patches are extracted with the Harris interest point detector from the training data and then clustered to generate the codebook. Additionally, for every codebook entry, its ground truth segmentation and the position it appeared in relative to the object centre is stored. During recognition, this information is used to perform a Hough transform to identify hypotheses for the object's centre. Given the object's centre and the ISM, a rough segmentation of the object can be found (see Fig. 2.12).



(a) (b) (c)

Figure 2.11: **An overview of the Fragment CRF system (Levin and Weiss, 2009)**. Fragments search for matches in the image (left). The fragments' local evidence (middle) guides the algorithm to make the final segmentation (right).

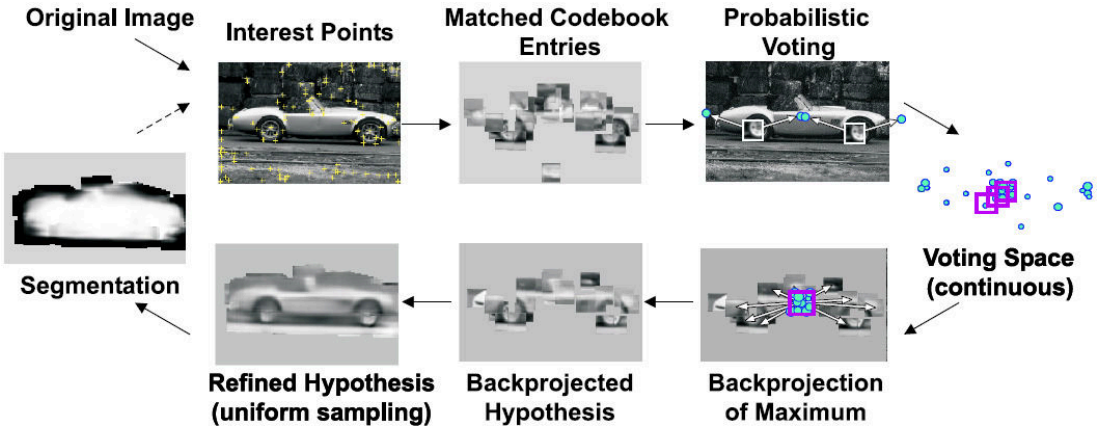


Figure 2.12: **Segmentation using an Implicit Shape Model (Leibe et al., 2004)**. Image patches are extracted around interest points and compared to the codebook. Matching patches then cast probabilistic votes, which lead to object hypotheses that can later be refined. Based on the refined hypotheses, a category specific segmentation is computed.

The main weakness of fragment-based methods is their reliance on extracted fragments or codebook entries for top-down segmentation. These methods require labelled data for training, and to obtain good segmentations of highly varying classes of objects a large number of fragments may be required. Additionally, finding the best cover of the object with the fragments stored in memory can often be computationally expensive.

2.3.2 Modelling the object's internal structure

2.3.2.1 Parts-based models

In this section we mention a collection of works that model *appearance* variability but impose rigid constraints on the shapes of the objects in the scene. Although such methods also reason about parts, they do so primarily to explain the variability of the object's appearance rather than that of its shape.

Jojic and Caspi (2004) capture image structure in *Probabilistic Index Maps* (PIMs). Given a palette, an index map assigns every pixel in the image a colour from that palette. Without changing the index map, one can arbitrarily change palettes to explain different images in a dataset (see Fig. 2.13).

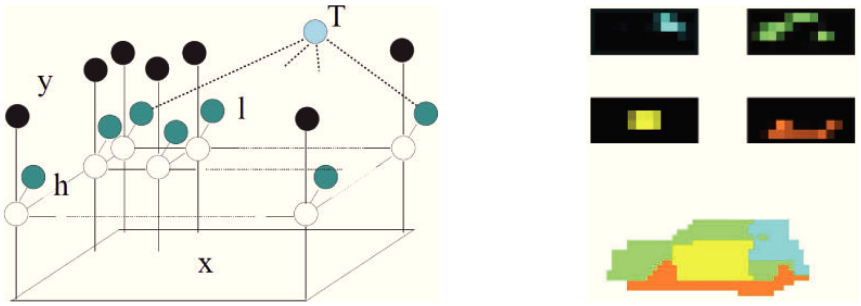
The Multiple Cause Vector Quantisation (MCVQ) model (Ross and Zemel, 2006) is a similar generative latent factor model in which each data dimension is allowed to select a different 'part' as its explanation. Whereas part appearance is specified by a uniform colour or texture in the PIM, it is represented by an exemplar mean image in the MCVQ. In the same work the authors also present Multiple Cause Factor Analysis (MCFA), which uses a Factor Analysis model for part appearances. This allows the model to learn to represent images using potentially larger and more meaningful parts.

A similar model is presented in Kannan et al. (2006), except now part descriptions are learned on a single 'jigsaw' image that allows for automatic information sharing between parts. Although robust to appearance variations, the constraints embedded into such models make them unsuitable for modelling images with structural variability.

With the Layout Consistent Random Field (LCRF, Winn and Shotton, 2006) and the Located Hidden Random Field (LHRF, Kapoor and Winn, 2006), the authors demonstrate how CRF models can be extended to learn about parts in an unsupervised fashion. In their hierarchical CRF framework, rather than modelling the interaction be-

(a) (b)

Figure 2.13: **The Probabilistic Index Map method (Jojic and Caspi, 2004)**. Given a set of input images (bottom), the model learns a distribution on index maps (a sample from which can be seen on the left) that captures the colour-invariant structure in the data. Latent 'palettes' (top) are inferred to explain the data. Labelled training data can be used in conjunction with the index maps to perform object segmentation.



(a) (b)

Figure 2.14: **The Located Hidden Random Field (Kapoor and Winn, 2006)**. (a) Graphical model for the LHRF. (b) The top row shows the probabilities of each part occurring within the object reference frame. The bottom row shows the corresponding part labellings of an example test image.

tween foreground and background labels through the pairwise potentials, they model the local interaction between part labels. They ensure that the parts are spatially localised relative to each other by introducing a discrete latent variable – representing the position of the object – that is connected to every label variable (see Fig. 2.14).

2.4 Explicit models of occlusion

We finally mention several works which explicitly focus on modelling *occlusion*. Even when the appearances and shapes of all objects in a scene are constant, their positions relative to the camera can have complex effects on the image's final pixel values.

An early example of this is Koller et al. (1994), where occlusion is explicitly dealt with to obtain more accurate trajectory estimates when tracking multiple cars on a highway. Williams and Titsias (2004) use greedy learning algorithms to efficiently explain sequences of images in terms of layered *sprites* – appearance images with associated masks – that translate and combine occlusively to generate the sequence. Note that in this work the assumption is that object appearances remain relatively constant throughout the dataset. In Occlusive Components Analysis (OCA, Lücke et al., 2009), a generative model (in which layer ordering is explicitly modelled as a latent random variable) is presented and approximate Expectation-Maximisation algorithms are derived for learning. Although the model can account for appearance variation (changes in hue) between images, object positions are assumed to be fixed. Le Roux et al. (2011) develop an alternative framework for reasoning about occlusion at the patch and image level, in which the shapes and appearances of each of the parts that compose the image are modelled separately. Although models such as these can be beneficial when training data is expected to include occlusions, the issue of occlusion is orthogonal to that of object representation – what the best model for object shape is in the first place.

2.5 Summary

In this chapter we have highlighted a number of existing approaches for probabilistic segmentation in the literature (see Table 2.1 and Fig. 2.15 for a summary). Most of the techniques incorporate low-level information in one form or another – typically with the use of random fields. Such models define probability distributions over the labels of *sites* (be it individual pixels, image patches or superpixels) that help the algorithms to resolve the ambiguities that arise when considering local evidence for sites in isolation.

However, such low-level statistics are often not enough. State-of-the-art techniques additionally incorporate higher-level information to guide segmentation. In most cases, these cues are derived from the model's *a priori* knowledge about the object's shape. This information is combined with low-level cues to obtain accurate segmentations.

The way in which shape is represented differs between models and the accuracy of these models is dependent on the degree to which their shape priors can match the object's outline in unseen images. In many cases, segmentation techniques only mainly differ in how well they represent and learn about the variability in the object's shape.

Model	Shape model	Appearance model	Parts	Continuity	Machinery	Training
LSM *Frey et al. (2003)*	Factor Analysis	Factor analysis	-	-	Generative	Unsupervised (video)
OBJCUT *Kumar et al. (2005)*	Pictorial Structure	Histograms	✓	CRF	Generative	Unsupervised (video)
LOCUS *Winn and Jojic (2005)*	Deformable mask	Gaussian mixture model	-	CRF	Generative	Unsupervised
DPMCRF *Larlus et al. (2009)*	Composition of blobs	SIFT, hue and location features	-	CRF	Generative	Unsupervised
SCA *Jojic et al. (2009)*	Deformable mask	Histograms or Gaussian mixture model	✓	-	Generative	Unsupervised
ClassCut *Alexe et al. (2010)*	Mean mask	RGB and SURF	-	CRF	Discriminative	Unsupervised
Fragments *Borenstein et al. (2004)*	Fragments	RGB and textures	✓	CRF	Discriminative	Supervised
ISM *Leibe et al. (2004)*	Fragments	Codebook of appearance patches	✓	-	Discriminative	Supervised
GrabCut *Rother et al. (2004)*	-	Gaussian mixture model	-	CRF	Discriminative	No learning
MoCRF *He et al. (2006)*	-	RGB	-	CRF	Discriminative	Supervised
LHRF *Kapoor and Winn (2006)*	Part biases	SIFT	✓	CRF	Discriminative	Supervised
TextonBoost *Shotton et al. (2006)*	-	Texture-layout, colour, location features	-	CRF	Discriminative	Supervised
LCRF *Winn and Shotton (2006)*	-	SIFT	-	CRF	Discriminative	Supervised
SPCRF *Fulkerson et al. (2009)*	-	RGB	-	CRF	Discriminative	Supervised
BBOX *Lempitsky et al. (2009)*	-	LUV and location features	-	CRF	Discriminative	No learning
FCRF *Levin and Weiss (2009)*	Fragments	RGB	✓	CRF	Discriminative	Supervised

Table 2.1: **A summary of characteristics for a number of prominent segmentation techniques**. Many of the techniques considered in this chapter do not provide end-to-end solutions to the segmentation problem and therefore have not been included in this table.

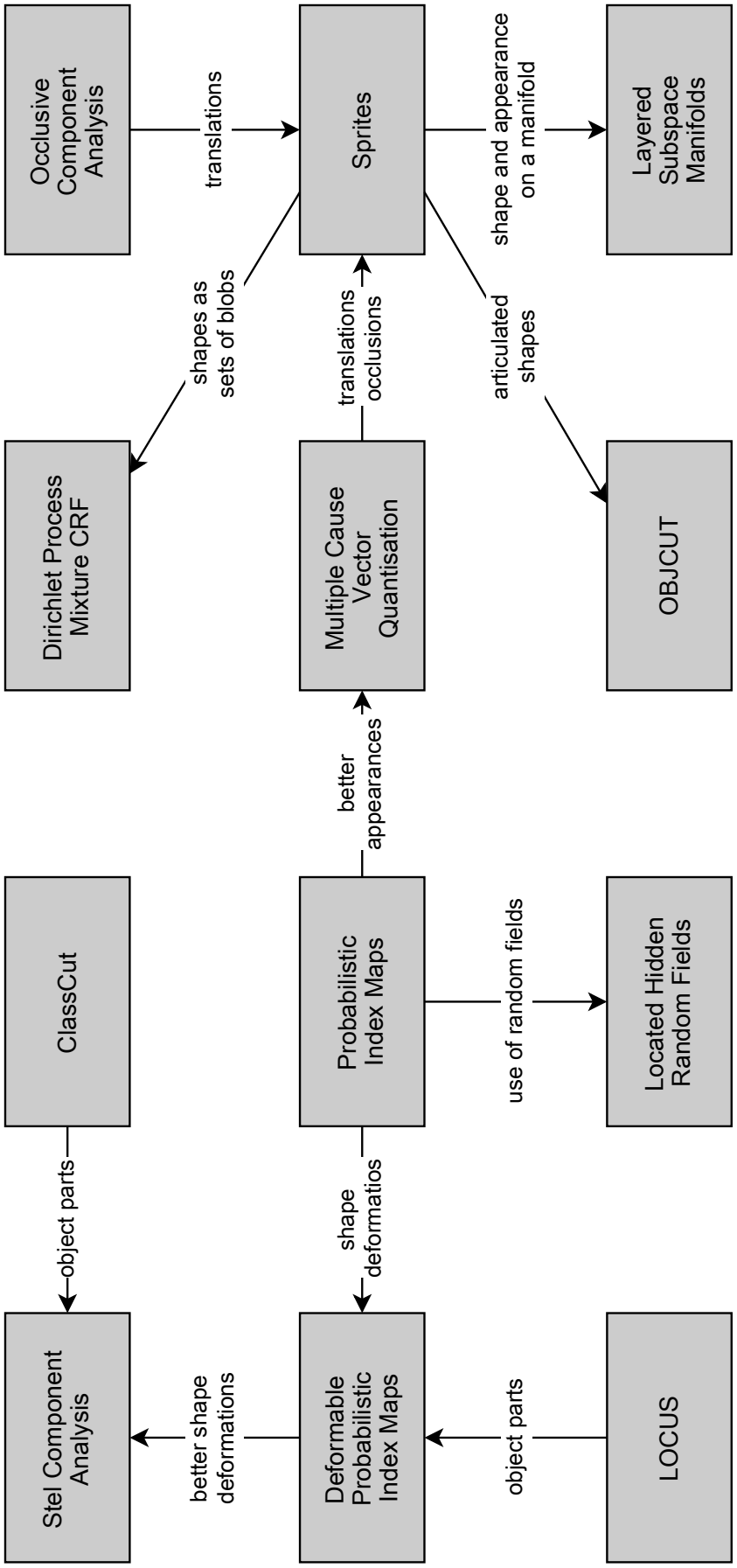


Figure 2.15: **One possible arrangement of relationships between a number of prominent generative segmentation techniques**.

Chapter 3

Combining Models of Part Shape and Appearance

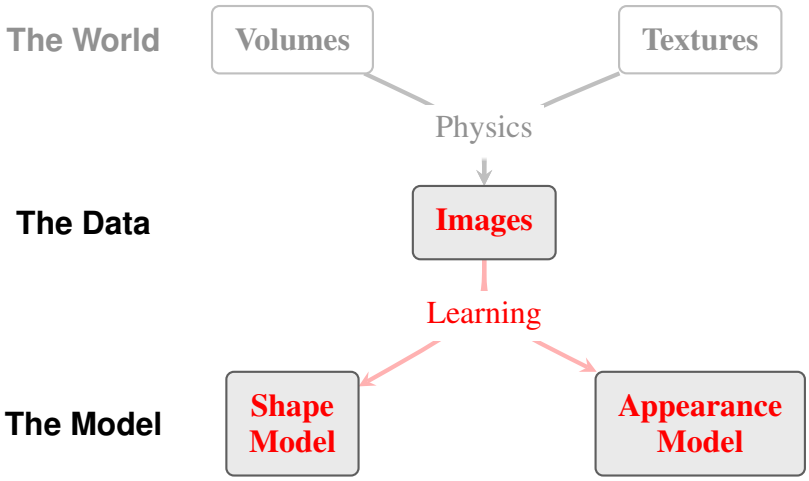


Figure 3.1: Learning shape and appearance models from unlabelled datasets.

There is a rich history of work on probabilistic models that segment by only considering low-level, pairwise pixel statistics (e.g. MRFs or CRFs; see Sec. 2.2). Errors in such models can typically be attributed to a lack of high-level, cross-image understanding about the object in question.

A number of models have been recently proposed that obtain more accurate segmentations by incorporating prior knowledge about the foreground object's shape. These probabilistic techniques mainly differ in how accurately they represent and learn about the variability in the object's shape (see Sec. 2.3 for a discussion of this issue).

In this chapter we present an explicit probabilistic model of images of objects based on a latent Gaussian model of shape, and show how it can be trained in an unsupervised fashion. We call this novel image representation *Factored Shapes and Appearances* (FSA). FSA is parts-based, and learns from datasets that exhibit variability in both shape and appearance.

Our experiments on a variety of datasets demonstrate the advantages of FSA's explicit modelling shape variability: First, we show that the model's latent representations can be interpreted as 'parsings' of images, and demonstrate that these parsings are accurate enough to be used for tasks like fine-grained categorisation. Second, we apply FSA to the object segmentation task, and show that its performance is comparable to that of the state-of-the-art on a number of benchmark datasets.

The remainder of this chapter is structured as follows: In Secs. 3.1 and 3.2 we present FSA and propose an efficient inference and learning scheme for the model. In Sec. 3.3 we provide a detailed explanation of how FSA generalises and extends previous work in the field. We provide an experimental evaluation of the model in Sec. 3.4 and conclude with a discussion in Sec. 3.5.

3.1 The FSA generative model

In FSA we consider datasets of images of an object class. We assume that the images are constructed through some combination of a fixed number of parts (which can alternatively be thought of as layers). Given a dataset $\mathbf{D} = \{\mathbf{X}^d\}$, $d = 1...n$ of such images $\mathbf{X}$, each consisting of P pixels $\{\mathbf{x}_i\}$, $i = 1...P$ in some feature space, we wish to infer a segmentation $\mathbf{S}$ for the image.

Each segmentation consists of a labelling $\mathbf{s}_i$ for every pixel, where L is the fixed number of parts that combine to generate the foreground and $\mathbf{s}_i$ is a 1-of-$(L+1)$ encoded variable. In other words, $\mathbf{s}_i = (s_{li})$, $l = 0...L$, $s_{li} \in \{0, 1\}$ and $\sum_l s_{li} = 1$. Note that the background is also treated as a 'part' ($l = 0$). Accurate inference of $\mathbf{S}$ is driven by FSA's models for 1) part shapes and 2) part appearances. In the following sections we describe how the two components are defined.

3.1.1 Shape

Let $\mathbf{m}_l$ be a collection of real numbers of the same size as the image, densely representing the model's preference for part l's shape at each location. These 'masks' combine via a softmax-like activation function to generate the segmentation $\mathbf{S}$. Let

$$\sigma_{li} = \frac{\exp\{\mathbf{m}_{li}\}}{\sum_{k=0}^{L} \exp\{\mathbf{m}_{ki}\}}, \tag{3.1}$$

then the distribution on the labelling of pixel i is given by

$$p(\mathbf{s}_{li} = 1|\boldsymbol{\theta}) = \epsilon + (1 - L\epsilon) \cdot \sigma_{li}. \tag{3.2}$$

Here ϵ is a 'leak' parameter that helps prevent over-confident predictions by 'smoothing out' the distribution imposed by the model on segmentations.

In order to be able to allow for part shape variability, the model is designed to capture a *distribution* over $\mathbf{m}_l, l = 1...L$ ($\mathbf{m}_0$ is fixed to equal $\mathbf{1}$). Specifically, the probability distribution over $\mathbf{m}_l$ is defined by a Factor Analysis-like model:

$$\mathbf{m}_l = \mathbf{F}_l\mathbf{v} + \mathbf{c}_l, \tag{3.3}$$

$$p(\mathbf{v}) = \mathcal{N}(\mathbf{0}, \mathbf{I}_{H\times H}). \tag{3.4}$$

Here $\mathbf{v}$ is an H-dimensional latent variable, $\mathbf{F}_l$ is a $D \times H$ matrix analogous to the *factor loading matrix* in Factor Analysis literature and $\mathbf{c}_l$ is the mean mask. An L^1-norm prior on $\mathbf{F}$ is used to reduce the amount of noise in its values.

We additionally consider an alternative shape variability model in which we use *separate*, $\bar{H}$ dimensional latent variables $\mathbf{v}_l$ for every part ($H = L \times \bar{H}$). This *local* model can be thought of as a special case of the *global* model presented earlier, in which most of the columns of each $\mathbf{F}_l$ are forced to equal 0. The local model is useful in cases where we believe the shapes of any two pairs of parts in the data to be independent (e.g. the pose of upper and lower parts of human bodies), and we wish to explicitly build this knowledge into the model. We explore the differences between the local and global models through several experiments in Sec. 3.4.1.

3.1.2 Appearance

Pixels corresponding to each part in a given image are assumed to have been generated by W fixed Gaussians in feature space (in our experiments we only use Lab colour

features, CIE (1978), but in general the space could contain any features including e.g. SIFT or SURF). In the pre-training phase, the means $\{\boldsymbol{\mu}_w\}$ and covariances $\{\boldsymbol{\Sigma}_w\}$ of these Gaussians are extracted by training a Gaussian mixture model with W components on every pixel in the dataset, ignoring image and part structure. It is also assumed that each of the L parts have different appearances in different images, and that these appearances can be clustered into K classes per part. The classes differ in how likely they are to use each of the W Gaussian components when 'colouring in' the part.

The generative process is as follows: For part l in a given image, one of the K classes is chosen (represented by a 1-of-K indicator variable $\mathbf{a}_l$). Given $\mathbf{a}_l$, the probability distribution defined on pixels associated with part l is given by a Gaussian mixture model with means $\{\boldsymbol{\mu}_w\}$ and covariances $\{\boldsymbol{\Sigma}_w\}$ and mixing proportions $\{\phi_{lkw}\}$. Therefore the distribution on the image pixel values is given by

$$p(\mathbf{x}_i|\mathbf{A},\mathbf{s}_i,\boldsymbol{\theta}) = \prod_{l=0}^{L} p(\mathbf{x}_i|\mathbf{a}_l,\boldsymbol{\theta})^{s_{li}} \tag{3.5}$$

$$= \prod_{l=0}^{L}\left(\prod_{k=1}^{K}\left(\sum_{w=1}^{W}\phi_{lkw}\,\mathcal{N}(\mathbf{x}_i|\boldsymbol{\mu}_w,\boldsymbol{\Sigma}_w)\right)^{a_{lk}}\right)^{s_{li}}. \tag{3.6}$$

The prior on $\mathbf{A} = \{\mathbf{a}_l\}$ specifies the probability of each appearance class being selected for the parts in any given image:

$$p(\mathbf{A}|\boldsymbol{\theta}) = \prod_{l=0}^{L} p(\mathbf{a}_l|\boldsymbol{\theta}) = \prod_{l=0}^{L}\prod_{k=1}^{K}(\pi_{lk})^{a_{lk}}. \tag{3.7}$$

See Fig. 3.2 for an illustration of the appearance model. In our experiments the model typically performs best when $K \simeq 10$, and $W \simeq 30$.

We additionally place a hyper-prior on $\boldsymbol{\phi}$ that is similar to the one presented in Brand (1999). Let

$$p(\boldsymbol{\phi}) \propto e^{-E(\boldsymbol{\phi})}, \tag{3.8}$$

where

$$E(\boldsymbol{\phi}) = \lambda_{\text{self}} \cdot E_{\text{self}}(\boldsymbol{\phi}) + \lambda_{\text{others}} \cdot E_{\text{others}}(\boldsymbol{\phi}), \tag{3.9}$$

$$E_{\text{self}}(\boldsymbol{\phi}) = -\sum_{l=0}^{L}\sum_{k=1}^{K}\sum_{w=1}^{W}\phi_{lkw}\cdot\ln\left(\phi_{lkw}\right), \tag{3.10}$$

$$E_{\text{others}}(\boldsymbol{\phi}) = -\sum_{l=0}^{L}\sum_{m\neq l}\left[\mathbf{D}_{\text{KL}}(\bar{\boldsymbol{\phi}}_l \parallel \bar{\boldsymbol{\phi}}_m) + D_{\text{KL}}(\bar{\boldsymbol{\phi}}_m \parallel \bar{\boldsymbol{\phi}}_l)\right]. \tag{3.11}$$

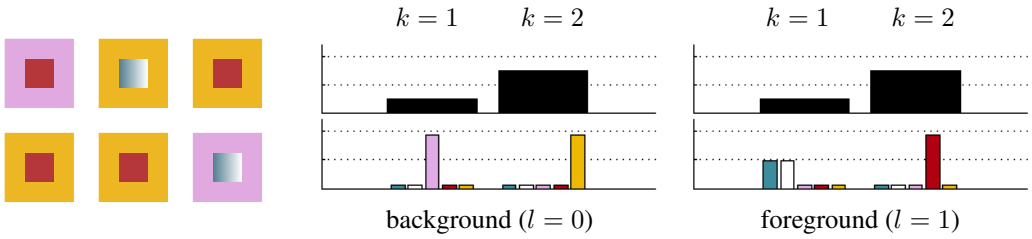


Figure 3.2: **Appearance modelling using mixtures of histograms**. Given a dataset of images and their segmentations, we construct a model of the parts' appearances. (a) An exemplar dataset. The foreground and background appear with 2 different styles. (b) The corresponding appearance model. The top row depicts $\boldsymbol{\pi}_l$ for the two parts and the bottom row depicts $\boldsymbol{\phi}_l$. In this example, the number of parts $L = 1$, the number of appearance classes $K = 2$ and the number of Gaussians $W = 5$.

Here

$$\bar{\phi}_{lw} = \sum_{k=1}^{K} \frac{\phi_{lkw}}{K} \tag{3.12}$$

is the average mixing proportion learned for part l and component w (this is of size 1×1), and

$$\bar{\boldsymbol{\phi}}_l = (\bar{\phi}_{lw}), \quad w \in \{0, ..., W\} \tag{3.13}$$

is the collection of average mixing proportions learned for part l for all components (this is of size $1 \times W$), and $D_{\mathrm{KL}}(\bar{\boldsymbol{\phi}}_l \parallel \bar{\boldsymbol{\phi}}_m)$ is the Kullback-Leibler divergence from the distribution defined by $\bar{\boldsymbol{\phi}}_l$ to that defined by $\bar{\boldsymbol{\phi}}_m$:

$$D_{\mathrm{KL}}(\bar{\boldsymbol{\phi}}_l \parallel \bar{\boldsymbol{\phi}}_m) = \sum_{w=1}^{W} \bar{\phi}_{lw} \cdot \ln\left(\frac{\bar{\phi}_{lw}}{\bar{\phi}_{mw}}\right). \tag{3.14}$$

This hyper-prior prefers settings of $\boldsymbol{\phi}$ that define

1. distributions on the appearance components for each class of each layer that are low-entropy (via E_{self}),
2. distributions on the appearance components for each of the layers that are dissimilar from each other (via E_{others}),

and can be very effective in accelerating convergence of the parameters during training. Suitable values of λ_{self} and λ_{others} are found through trial and error.

3.1.3 Occlusion

Instead of modelling part occlusion using an explicit random variable, FSA captures knowledge about part-ordering *implicitly* in the shape parameters. By increasing the magnitude of m_{li} for a particular l, the model can capture the increased likelihood of part l occluding other parts at pixel i. In cases where the multiple parts are equally likely to occlude each other, the appearance model is used to resolve this ambiguity in the posterior. See Fig. 3.3 for an illustration of this effect.

3.1.4 Combined model

To summarise, the latent variables $\mathbf{Z}$ for image $\mathbf{X}$ are $\mathbf{A}$, $\mathbf{S}$ and $\mathbf{v}$, the model's active parameters $\boldsymbol{\theta}$ include shape parameters $\boldsymbol{\theta}^s = \{\{\mathbf{F}_l\}, \{\mathbf{c}_l\}\}$ and appearance parameters $\boldsymbol{\theta}^a = \{\{\pi_{lk}\}, \{\phi_{lkw}\}\}$, and

$$p(\mathbf{X}, \mathbf{A}, \mathbf{S}, \mathbf{v}|\boldsymbol{\theta}) = p(\mathbf{v})\, p(\mathbf{A}|\boldsymbol{\theta}^a) \prod_{i=1}^{P} p(\mathbf{s}_i|\mathbf{v}, \boldsymbol{\theta}^s)\, p(\mathbf{x}_i|\mathbf{A}, \mathbf{s}_i, \boldsymbol{\theta}^a). \tag{3.15}$$

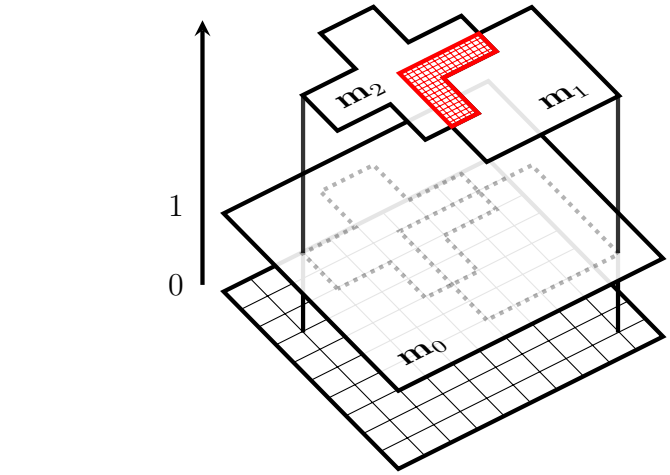
See Fig. 3.4 for an illustration of the complete FSA graphical model. During learning, we find the values of $\boldsymbol{\theta}$ that maximise the likelihood of the training data $\mathbf{D}$, and segmentation is performed on previously-unseen image by querying the marginal distribution $p(\mathbf{S}|\mathbf{X}^{\text{test}}, \boldsymbol{\theta})$.

3.2 Inference and learning

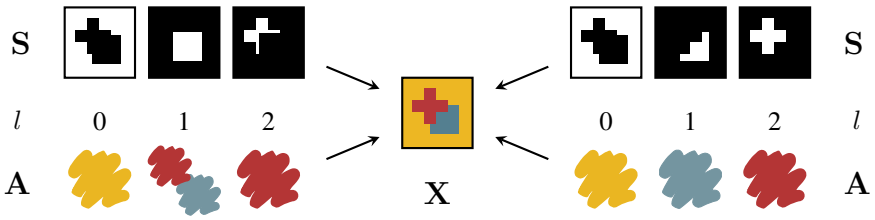
We use the expectation-maximisation (EM) algorithm to find estimates of the maximum likelihood parameters.

For the E-step, we wish to find $p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}) = p(\mathbf{A}, \mathbf{S}, \mathbf{v}|\mathbf{X}, \boldsymbol{\theta})$. However, the exact evaluation of this distribution is intractable. Instead we approximate $p(\mathbf{A}, \mathbf{S}, \mathbf{v}|\mathbf{X}, \boldsymbol{\theta})$ by drawing samples of $\mathbf{A}$, $\mathbf{S}$ and $\mathbf{v}$ using block-Gibbs Markov Chain Monte Carlo (MCMC):

$$p(\mathbf{A}, \mathbf{S}, \mathbf{v}|\mathbf{X}, \boldsymbol{\theta}) \simeq \frac{1}{T} \sum_{t=1}^{T} \delta(\mathbf{A}^{(t)}, \mathbf{S}^{(t)}, \mathbf{v}^{(t)}). \tag{3.16}$$

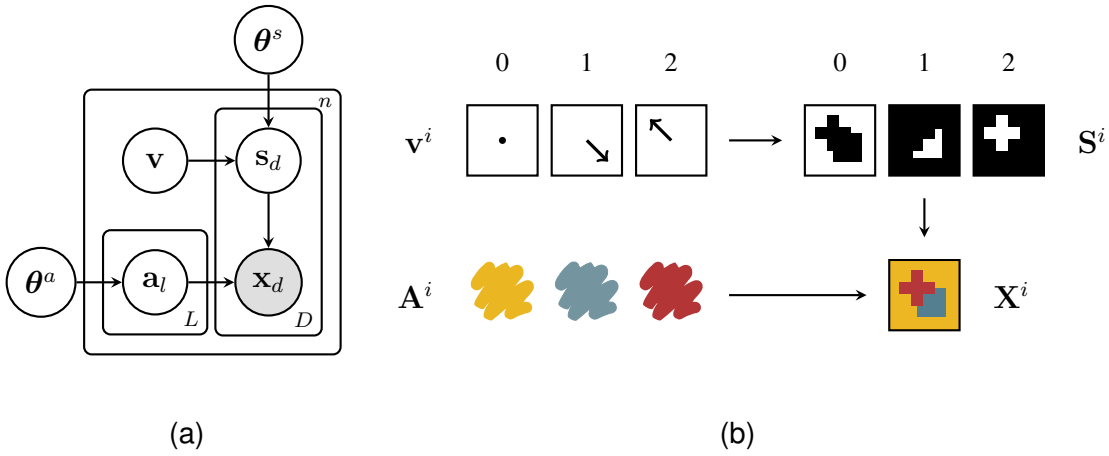


(a)



(b)

Figure 3.3: **Lazy occlusion reasoning**. (a) Given the image $\mathbf{X}$, the masks are deformed to their most likely states. In this example, the model has learned that the cross and square always appear in front of the background and that they are equally likely to be in the foreground. The highlighted pixels (red) are equally likely to belong to either shape at this stage. (b) *Left:* One setting of $\mathbf{A}$ and $\mathbf{S}$ that can explain $\mathbf{X}$. Note the two-tone appearance for part 1. *Right:* The most likely setting of $\mathbf{A}$ and $\mathbf{S}$. Out of all such competing segmentations, the most likely $\mathbf{S}$ is the one for which the corresponding choice of appearances is most probable.



(a) (b)

Figure 3.4: **The Factored Shapes and Appearances model**. (a) Directed graphical representation of the global FSA model. Pixel intensities $\mathbf{x}_i$ are modelled via L appearance random variables ($\mathbf{a}_l$). The model's belief about each part's shape is captured by a latent variable ($\mathbf{v}$). Here $\mathbf{v}$ represents the slight translation of the cross and square, but in theory it can represent any kind of deformation. Segmentation random variables ($\mathbf{s}_i$) assign each image pixel to a part. (b) Schematic diagram of the model for a single image $\mathbf{X}^d$.

where $\delta(x)$ is the Dirac delta 'function' centred at x and T is the number of samples drawn after some burn-in period.

The appearance variable $\mathbf{A}^d$ is sampled given the d-th image $\mathbf{X}^d$ and its corresponding segmentation $\mathbf{S}^d$. The conditional distribution of appearance class k being chosen for part l (i.e. the binary variable a_{lk} being set to 1) is given by

$$p(a_{lk}=1|\mathbf{S},\mathbf{v},\mathbf{X},\boldsymbol{\theta}) = \frac{\pi_{lk}\prod_{i=1}^{D}\left(\sum_{w=1}^{W}\phi_{lkw}\,\mathcal{N}(\mathbf{x}_i|\boldsymbol{\mu}_w,\boldsymbol{\Sigma}_w)\right)^{s_{li}}}{\sum_{r=1}^{K}\left[\pi_{lr}\prod_{i=1}^{D}\left(\sum_{w=1}^{W}\phi_{lrw}\,\mathcal{N}(\mathbf{x}_i|\boldsymbol{\mu}_w,\boldsymbol{\Sigma}_w)\right)^{s_{li}}\right]}. \tag{3.17}$$

The segmentation variable $\mathbf{S}^d$ is then sampled given $\mathbf{v}^d$ and $\mathbf{A}^d$. It can be shown that the conditional distribution of the segmentation factorises over the pixels in the image. The probability of pixel i being associated with part l is

$$p(s_{li}=1|\mathbf{A},\mathbf{v},\mathbf{X},\boldsymbol{\theta}) = \frac{p(s_{li}=1|\mathbf{v},\boldsymbol{\theta})\,p(\mathbf{x}_i|\mathbf{A},\mathbf{s}_i)}{\sum_{m=1}^{L}p(s_{mi}=1|\mathbf{v},\boldsymbol{\theta})\,p(\mathbf{x}_i|\mathbf{A},\mathbf{s}_i)}. \tag{3.18}$$

Finally, $\mathbf{v}^d$ is sampled given the segmentation $\mathbf{S}^d$. To do this we use an efficient elliptical slice sampling scheme (Murray et al., 2010). In each iteration of the top-level

block-Gibbs sampler, the sample for $\mathbf{v}^d$ is set to equal the mean of the samples returned by the elliptical slice sampler after a burn-in period. For detailed derivations of the sampling equations see Sec. A.1.

For the M-step we are looking to find $\arg\max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}})$, where

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}}) = \ln p(\boldsymbol{\theta}) + \sum_{d=1}^{n} \sum_{\mathbf{Z}^d} p(\mathbf{Z}^d | \mathbf{X}^d, \boldsymbol{\theta}^{\text{old}}) \ln p(\mathbf{X}^d, \mathbf{Z}^d | \boldsymbol{\theta}). \tag{3.19}$$

To do this, we compute the derivative of Q with respect to $\boldsymbol{\theta}^a$ and $\boldsymbol{\theta}^s$. The gradients are used in a numerical optimisation routine to find the settings of the parameters at which Q is maximised. We use independent scaled conjugate gradients (SCG) routines to update the shape and appearance parameters. Note that special care needs to be taken to ensure that the $\boldsymbol{\pi}$ and $\boldsymbol{\phi}$ variables sum to 1. We re-parametrise the model such that

$$\pi_{lk} = \frac{\exp\{\alpha_{lk}\}}{\sum_{r=1}^{K} \exp\{\alpha_{lr}\}} \tag{3.20}$$

and

$$\phi_{lkw} = \frac{\exp\{\beta_{lkw}\}}{\sum_{q=1}^{W} \exp\{\beta_{lkq}\}}, \tag{3.21}$$

and optimise Q with respect to $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ instead. For detailed derivations of the gradient updates see Sec. A.2.

3.3 Related work

In Chapter 2 we surveyed a broad range of segmentation techniques at a high level; here we provide a detailed comparison of FSA with several closely related parts-based image models.

Existing parts-based image models can be categorised by the amount of variability they expect to encounter in the data and by how they model this variability. For example, in the Layered Subspace Manifold (LSM, Frey et al., 2003) *videos* are partitioned into layers that translate independently of each other. The layers exhibit limited shape and appearance variability from frame to frame, which is modelled using Factor Analysers and a fixed, explicit occlusion ordering. With the Sprites model, Williams and

Titsias (2004) show how such layered models can be efficiently learned one layer at a time, however they do not model shape or appearance variability. By contrast, FSA is designed to work on datasets of *images* that exhibit significant shape and appearance variability from image to image, and does not impose any layer ordering into the model.

With Multiple Cause Vector Quantisation (MCVQ), Ross and Zemel (2006) present an alternative part-based representation of images. The model learns a mean prior over the partitioning of the image, and it is assumed that a fixed number of appearance templates generate the pixels within each part. When applied to highly variable data, the model may find it difficult to learn meaningful parts as it can only make limited variations in the partitionings from image to image. The authors also present Multiple Cause Factor Analysis (MCFA), which uses a Factor Analysis (FA) model for part appearances, however FA learns distributions over part appearances that are too restrictive for most datasets of interest. By contrast, FSA explicitly models the variability of pixel assignments to parts, therefore learning sharp partitions, and it models part appearance variation using histograms which can be more flexible than a mean or an FA model.

Heess et al. (2011) propose an extension of the Restricted Boltzmann Machine that allows the joint shape and appearance of foreground objects to be learned. Like FSA the model is generative and is applied to the object segmentation problem, however it models the foreground with only a single part and therefore is not expected to cope with the variability present in the most challenging datasets. In Le Roux et al. (2011) a similar framework is used to model images with multiple objects (which could be considered to be analogous to 'parts' used here) however no attempt is made to model the interactions between said objects. Part shapes are likely to vary in concert with each other, and it would therefore be desirable to explicitly model such interactions.

We also mention the work of Sudderth and Jordan (2008) who use a non-parametric prior over the number of parts (in their terminology 'segments'), and use thresholded Gaussian Processes to model each part. They employ this model to segment images of natural scenes. FSA is similar in that each part has an associated 'activation surface' (the $\mathbf{m}_l$ masks) except that in FSA these activation surfaces incorporate prior information about the parts in question.

The closest works to ours are LOCUS (Winn and Jojic, 2005) and Stel Component

Analysis (SCA, Jojic et al., 2009). In the basic formulation of LOCUS, the model uses only one 'part' to account for the foreground object, but this restriction can be relaxed with the deformable probabilistic index map (dPIM, Winn and Jojic, 2005). Shape variability between images is accounted for using a deformation field that warps the partitioning to fit each image. Since the formulation imposes only local smoothness constraints on deformations, samples from the model in the absence of an image are unlikely to capture global properties of the object in consideration (e.g. pose of a horse).

The SCA model, on the other hand, accounts for shape variability by learning a fixed number of templates for each part. The templates are restricted such that any pixelwise, convex combination of templates results in a valid probabilistic index map (i.e. one in which the probabilities of part assignments for each pixel sum to 1). The SCA distribution over segmentations is accurate only in the posterior – in the absence of an image, the defined distribution over segmentations is 'blurry'. Thus samples of partitionings generated by LOCUS and SCA will not have much resemblance to their training images, even though the are both generative models of image partitionings.

In FSA part shapes vary accurately even *in the prior* and segmentations randomly sampled by the model are similar to those found in the training data. Additionally, both LOCUS (with dPIMs) and SCA define global distributions over partitionings that do not factorise over part shape. In FSA parts can be modelled independently of each other allowing further developments to be made by incorporating specialised part models that concentrate on the shape, position and scale of each individually.

3.4 Experiments

FSA, as a generative model for images of objects, can be used to accomplish a variety of tasks in computer vision. FSA segments all images across the dataset simultaneously to learn a parts-based object model. In addition to the segmentations made by the algorithm, we inspect the parameters learned by the model. We show that these parameters form an intuitive reflection of the algorithm's 'understanding' of the object class.

We first illustrate the way in which FSA learns models of shape and appearance by considering synthetic datasets in Sec. 3.4.1. We then examine the model's behaviour

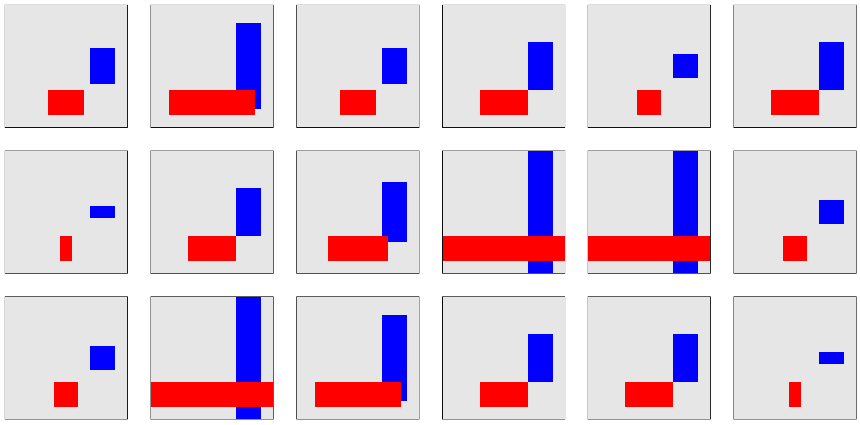


Figure 3.5: **A subset of the synthetic training images**. In each image the two bars are synchronised (i.e. their lengths are equal), and the red bar always occludes the blue bar.

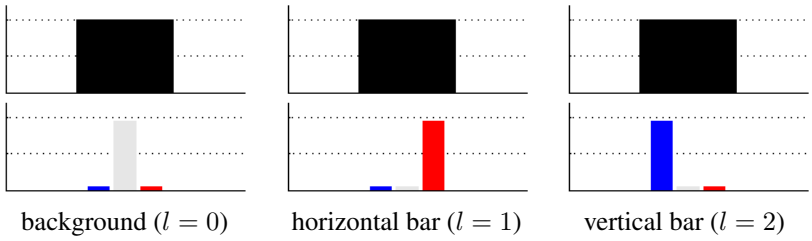


Figure 3.6: **The learned appearance model**. Here the number of parts $L = 2$, the number of appearance classes $K = 1$ and the number of Gaussians $W = 3$. The background is always grey, the horizontal bar is always red and the vertical bar is always blue.

on several real datasets in Secs. 3.4.2 and 3.4.3. Finally, we demonstrate how FSA can be used for the object segmentation task in Sec. 3.4.4.

3.4.1 Synthetic data

Consider a synthetic dataset of two bars of variable length such as the one shown in Fig. 3.5. Notice that in each image the two bars are synchronised (i.e. their lengths are equal), and that the red bar always occludes the blue bar. We train a global FSA model ($L = 2$, $H = 1$) on this dataset.

First we plot the learned appearance model upon convergence of the learning algorithm in Fig. 3.6. In Fig. 3.7 we plot samples drawn from the learned shape model. By doing

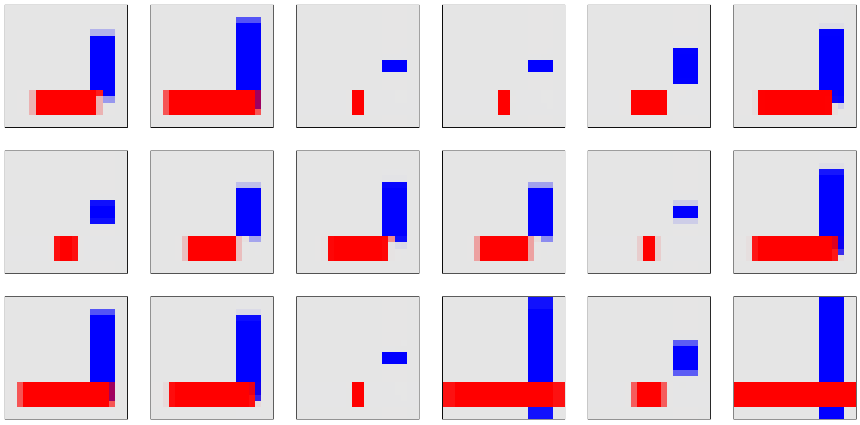


Figure 3.7: **Random samples from the learned global model**. Here we consider a global model with $H = 1$. The two bars are always of the same length. Pixel intensities are also sampled from the learned appearance model.

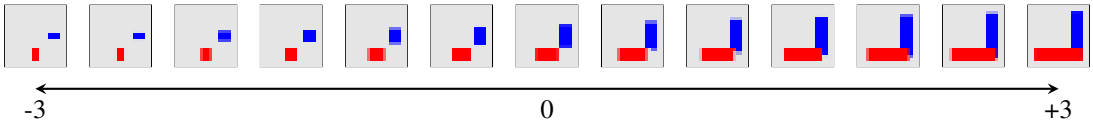


Figure 3.8: **The image structure varies as $\mathbf{v}$ moves in 1D space**. Here we consider a global model with $H = 1$. The two bars are always of the same length. Pixel intensities are also sampled from the learned appearance model.

so we get an indication of the kinds of shapes the model deems likely. Notice how the bars vary in length, are always of the same size, and appear with the correct occlusion ordering. It is also informative to inspect how the latent $\mathbf{v}$ variable is projected by $\mathbf{F}_l$ and $\mathbf{c}_l$ into masks for the parts. In Fig. 3.8 we show the way in which the image structure varies as $\mathbf{v}$ moves in 1D space.

We also train a *local* FSA model ($\bar{H} = 1$ per layer), and plot the samples it generates in Fig. 3.9. The generated bars now appear with different lengths in the same image. The model has *generalised* from the training data. Notice how the blue bar is ragged at its tip. This is to be expected, since the model has never actually 'seen' what the tip should look like in the training data – it has always been occluded by the red bar in that region.

We train the same local FSA model again, but this time on an unsynchronised dataset in which the two bars appear with different lengths in each image. We plot samples from the model in Fig. 3.10, and in Fig. 3.11 we show the way in which the model's

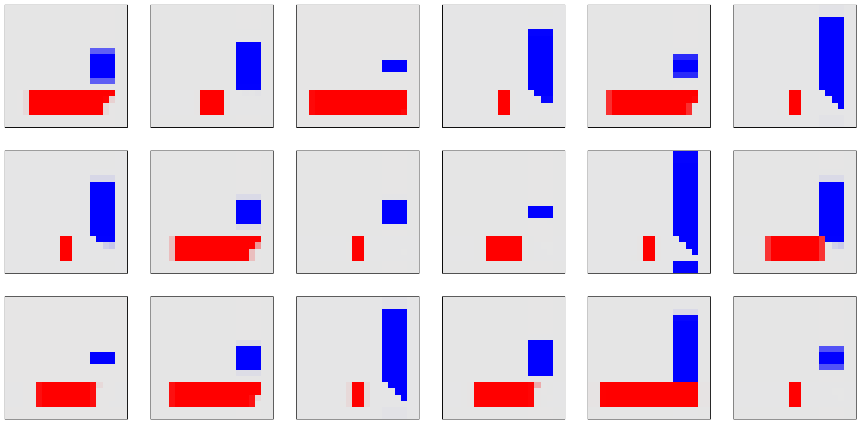


Figure 3.9: **Random samples from the learned local model**. Here we consider a local model with $\bar{H} = 1$. Synchronised training data. The two bars are of varying lengths. The blue bar is at times ragged. Pixel intensities are also sampled from the learned appearance model.

distribution on image structure varies as $\mathbf{v}$ moves in 2D space.

These results demonstrate the way in which FSA learns about the shape variability observed in the data. The global FSA model faithfully captures the covariance of the two bars, and the local model can be used to force the model to generalise.

3.4.2 Cars dataset

The first real dataset we consider contains 20 images of cars that have been downloaded from a manufacturer's website[1]. In addition to appearance variability, the cars exhibit significant shape variability across the dataset (e.g. hatchback, SUV, coupé, saloon, estate). After training on the dataset, the model's inference of $\mathbf{S}$ for an image $\mathbf{X}$ can be interpreted as a segmentation of that image. The segmentations inferred by an *unsupervised* FSA model with $L = 3$ and $H = 2$ are shown in Fig. 3.12. First, note that the model learns parts that appear to have some semantic meaning, despite the complete lack of supervision (cyan = body; yellow = glass, wheels and attached shadow; red = light background; navy = dark background). Also note that due to the fact that it incorporates a shape model, FSA produces accurate segmentations even when the car's appearance is similar to that of the background.

It will again be informative to inspect how the latent $\mathbf{v}$ variable is projected by $\mathbf{F}_l$ and

[1] http://bmw.com

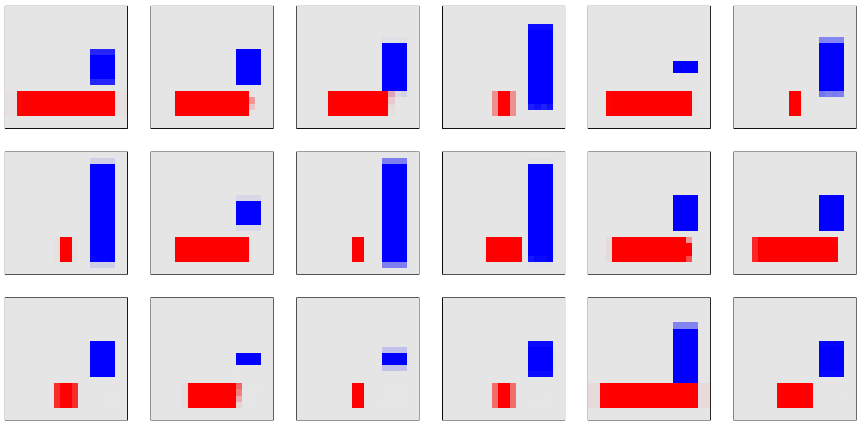


Figure 3.10: **Random samples from the learned local model**. Here we consider a local model with $\bar{H} = 1$. Unsynchronised training data. The blue bar is no longer ragged. Pixel intensities are also sampled from the learned appearance model.

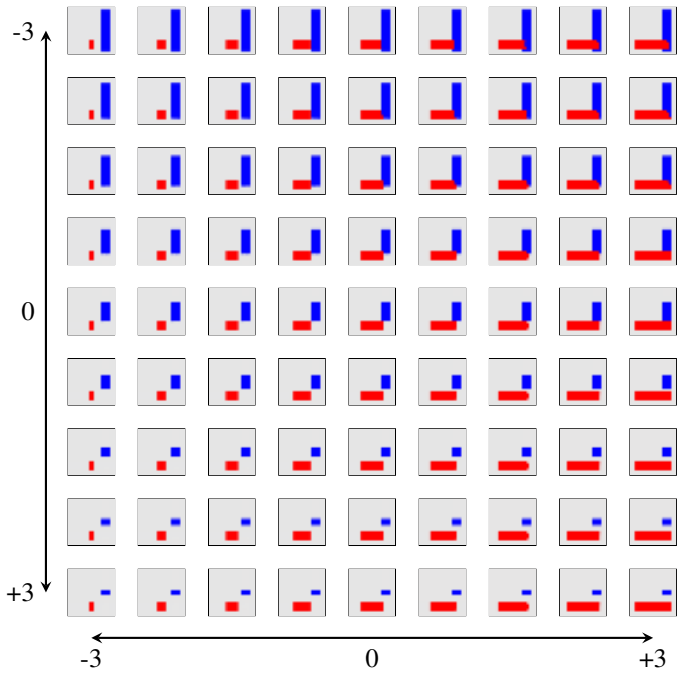


Figure 3.11: **The image structure varies as $\mathbf{v}$ moves in 2D space**. Here we consider a local model with $\bar{H} = 1$. Unsynchronised training data. Pixel intensities are also sampled from the learned appearance model.

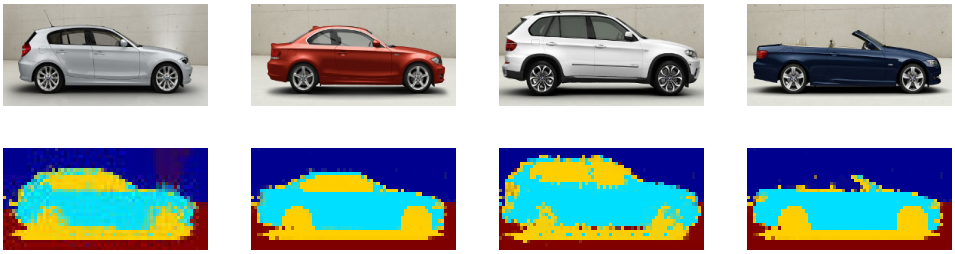


Figure 3.12: **Results on the BMW cars dataset**. A subset of the training images with their inferred segmentations. Distinct colours indicate assignments of pixels to different parts. The apparent 'blurriness' of samples is *not* due to averaging or resizing. We display the *probability* of each pixel belonging to different parts. If, for example, there is a 50-50 chance that a pixel belongs to the red or blue parts, we display that pixel in purple.

$\mathbf{c}_l$ into masks for the parts. In Fig. 3.13 we plot columns of one of the $\mathbf{F}$ matrices, and in Fig. 3.14 we plot the car body's mask for a grid of $\mathbf{v}$ values in 2-dimensional latent space. Notice how FSA learns a model of shape that gradually morphs between the parts' possible outlines. In doing so it learns a model of object class shape that is more informative than just a mean FSA model on the same dataset. Also note that the model learns a mask for the roof-less 'coupé' body type. A deformation field like the one used in LOCUS (Winn and Jojic, 2005) would find this kind of variability difficult to represent.

Finally, we observe that the inferred $\mathbf{v}$s can be used as discriminative indicators of the object's *type*. To test this, we labelled the images into 5 sub-categories (hatchback, SUV, coupé, saloon and estate). We then trained out-of-the-box SVM classifiers on *only* the inferred $\mathbf{v}$s in a leave-one-out scheme (first train on all available images except one, then test the resulting classifier on the image that was left out, and repeat this experiment until all images have been tested on precisely once), and found that the inferred $\mathbf{v}$s were sufficiently expressive to obtain 100% classification accuracy. We take a closer look at FSA's potential for fine-grained categorisation in Appendix B.

3.4.3 Other datasets

We apply the FSA model to a number of other datasets including 100 MIT pedestrians (Oren et al., 1997), 200 UMIST faces (Graham and Allinson, 1998) and 127 Caltech

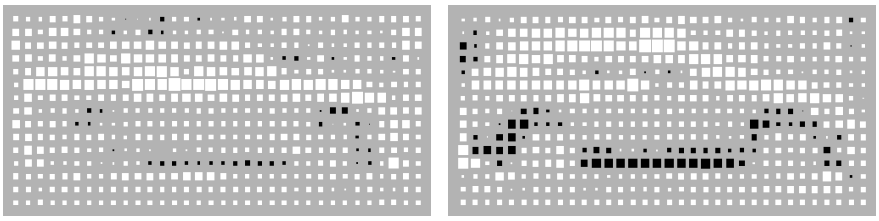


Figure 3.13: **Results on the BMW cars dataset**. Hinton diagrams of the two columns of $\mathbf{F}_2$ corresponding to the car body (cyan).

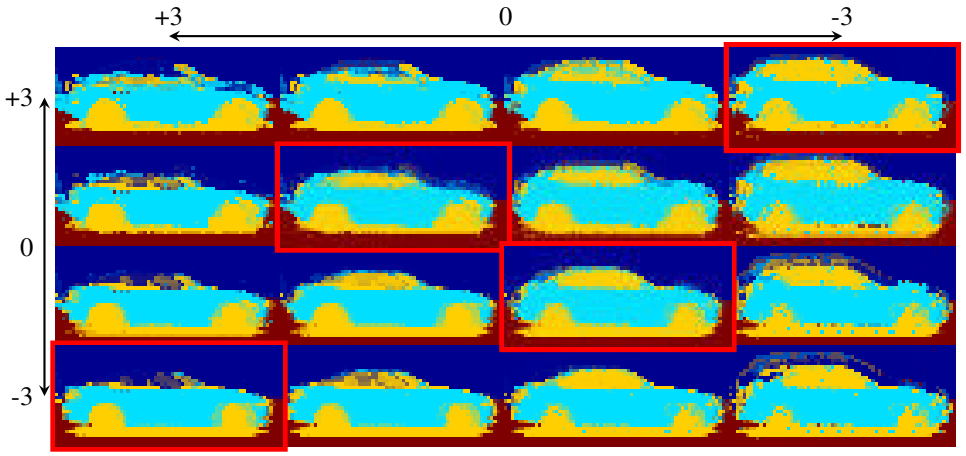


Figure 3.14: **Results on the BMW cars dataset**. A plot of the joint segmentation for a grid of $\mathbf{v}$ values in 2D latent space. Prototypical shapes of 4 of the different car types have been highlighted in red.

motorbikes (Fei-Fei et al., 2004), as well as 138 images of dresses obtained from a fashion retailer's website[2]. We use the following model parameters for the different datasets: pedestrians: $L = 3$, $H = 2$; faces: $L = 2$, $H = 2$; motorbikes: $L = 3$, $H = 20$; dresses: $L = 1$, $H = 5$. The results of these experiments can be seen in Fig. 3.15.

The model does a good job of learning about class shape across the dataset. In our experiments we observed that it uses this information effectively to guide inferences for more difficult images that cannot be segmented based on appearance cues alone.

Crucially, the fact that it has the flexibility to learn about shape deformations increases its chances of transferring shape information in a useful way. For example, having correctly learned about the shape of a human in an unusual pose in an image with

[2] http://marksandspencer.com

strong appearance cues, the model uses this information to correctly segment more difficult images of humans with the same pose. The *mean* pose in this case would do more harm than good in providing cues for segmentation.

3.4.4 Weizmann horses and Caltech4

We additionally evaluate the performance of the FSA model at segmenting the Weizmann horse (Borenstein et al., 2004) and Caltech4 (Fergus et al., 2003) datasets, where the ground truths are readily available. The train-test split for the datasets were as follows: Weizmann horses: 127-200; Caltech cars: 63-60; faces: 335-100; motorbikes 698-100 and airplanes: 700-100.

In supervised FSA, training is performed given the ground-truth segmentations for each image ($L = 1$). By contrast, in unsupervised FSA, no extra data is provided other than to manually assign the model's L parts to either the foreground or the background.

The baseline we consider is the batch GrabCut algorithm described by Alexe et al. (2010). GrabCut is initialised by training a foreground colour model on the central 25% of each test image and a background colour model using the remainder of its pixels.

The results of these experiments can be seen in Table 3.1. For comparison we also include accuracies reported by Borenstein et al. (2004, supervised), Winn and Jojic (2005, unsupervised) and Alexe et al. (2010, unsupervised). The discrepancy with LOCUS and Borenstein *et al.*'s approach on the Weizmann dataset is likely due to the lack of low-level edge features in our implementation of FSA. Supervised FSA outperforms the other models on the face and motorbike datasets, in part due to the way in which it learns to classify pixels belonging to necks as background and motorbike spokes as foreground.

3.5 Discussion

In this chapter we have presented a novel probabilistic model of objects that learns about shape and appearance by simultaneously segmenting all images in an unlabelled training dataset. The model is parts-based and factorial: if desired each of the parts can be modelled independently of the others. The model's descriptors for shape and

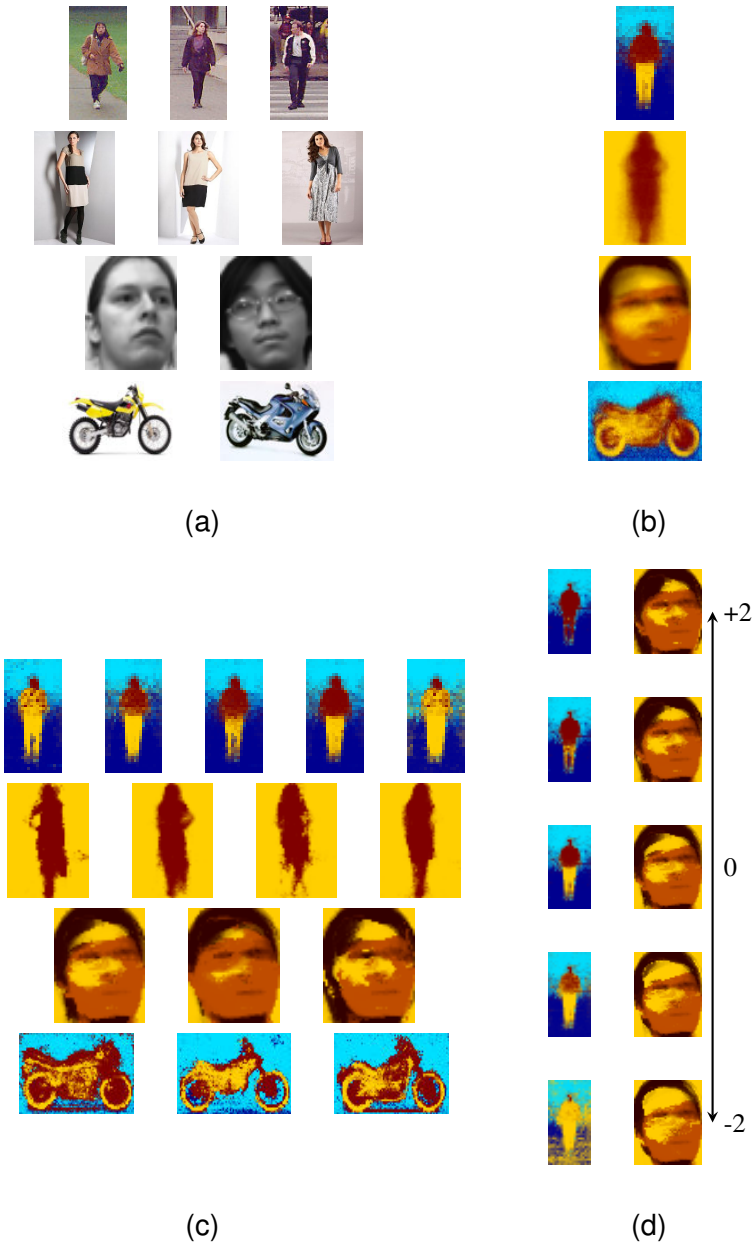


Figure 3.15: **Results on other datasets**. (a) Training images. (b) Partitioning learned by an FSA model with no shape deformation component (equivalent to a PIM). Distinct colours indicate probabilities of assignments of pixels to different parts. (c) A selection of samples from complete FSA models. Notice in row 1) captured variability of clothing styles and leg separation, 2) body poses, 3) face highlights and hair styles, and 4) motorcycle types. (d) Samples from the FSA pedestrian and face models as $\mathbf{v}$ moves on a 1D line in latent space. Notice how, for example, $\mathbf{v}$ affects the size of the forehead and the length of the hair.

Table 3.1: **Average segmentation accuracies**. Here we report the accuracy of the algorithm as the average percentage of correctly labelled pixels across all the test images.

	Weizmann		**Caltech4**		
	Horses	Cars	Faces	Motorbikes	Airplanes
GrabCut *Alexe et al. (2010)*	83.9%	45.1%	83.7%	82.4%	84.5%
Combined *Borenstein et al. (2004)*	**93.6%**	-	-	-	-
LOCUS *Winn and Jojic (2005)*	93.1%	91.4%	-	-	-
Arora *et al.* *Arora et al. (2007)*	-	**95.1%**	92.4%	83.1%	**93.1%**
ClassCut *Alexe et al. (2010)*	86.2%	93.1%	89.0%	90.3%	89.8%
Unsupervised FSA	87.3%	82.9%	88.3%	85.7%	88.7%
Supervised FSA	88.0%	93.6%	**93.3%**	**92.1%**	90.9%

appearance are particularly well suited to highly variable datasets of images. We have demonstrated that FSA can learn accurate models of shapes and appearances across a range of datasets, and that its latent representation can be used to accomplish a variety of common computer vision tasks, including object segmentation.

In Chapter 4 we take a closer look at the properties of the latent Gaussian shape model used in FSA, and ask if it is possible to devise shape models that address its drawbacks.

In Appendix B we present initials results showing how FSA's latent representation of part shape can be used for the fine-grained visual categorisation task, where the goal is to distinguish between, e.g., species of animals and plants or car and motorcycle types. Our initial results look promising and we believe this to be a potentially fruitful avenue for future research.

Chapter 4

Modelling Object Shapes

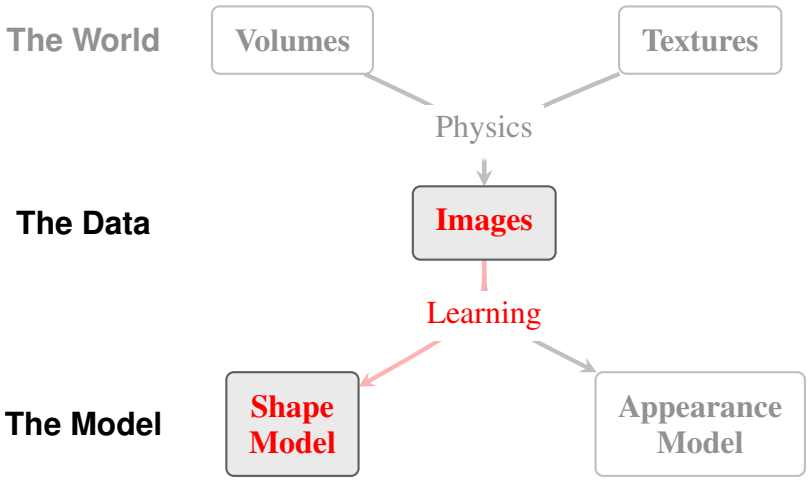
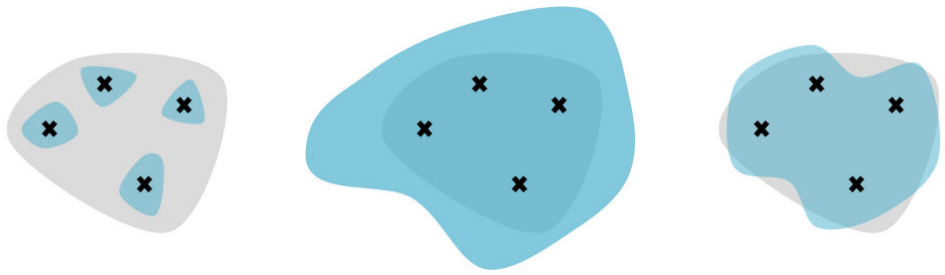


Figure 4.1: Learning a state-of-the-art model of shape.

Models of the shape of an object play a crucial role in many imaging algorithms, such as those for object detection and segmentation (e.g. the FSA model presented in Chapter 3, also Borenstein et al., 2004; Winn and Jojic, 2005; Alexe et al., 2010), inpainting (e.g. Chan and Shen, 2001; Bertozzi et al., 2007; Shekhovtsov et al., 2012) and graphics (e.g. Anguelov et al., 2005).

In object segmentation, local constraints on the shape, such as smoothness and continuity, can help provide correct segmentations where the object boundary is noisy or lost in shadow. More global constraints, such as ensuring the correct number of parts (legs, wheels, etc.), can resolve ambiguities where background regions look similar to an object part (e.g. Jojic et al., 2009).



(a) (b) (c)

Figure 4.2: **Realism vs. Generalisation**. Given the training data (black crosses) we wish to define a model distribution (blue) which is as close as possible to the underlying distribution of the data (gray). (a) A model that has overfit. It allocates probability mass only to correct regions of space, and therefore is realistic, but it also misses many important regions. (b) A model that has generalised too much. (c) A model that has generalised but is still realistic.

Shape also plays an important role in generative models of images (e.g. the FSA model presented in Chapter 3, also Frey et al., 2003; Williams and Titsias, 2004; Le Roux et al., 2011). In general, the better the model of object shape, the more performance should be improved in these applications.

In this chapter we address the question of how to build a 'strong' probabilistic model of object shapes. We define a strong model as one which meets two requirements:

1. **Realism** – samples from the model look realistic;
2. **Generalisation** – the model can generate samples that differ from training examples.

The first constraint ensures that the model captures shape characteristics at all spatial scales well enough to place probability mass only on images that belong to the 'true' shape distribution. The second constraint ensures that there are no gaps in the learned distribution, i.e. that it also covers novel unseen but valid shapes. See Fig. 4.2 for an illustration.

There have been a wide variety of approaches to modelling 2D shape in the literature. The most commonly used models are grid-structured Markov random fields or conditional random fields (MRFs and CRFs respectively, see Chapter 2). In such models, the pairwise potentials connecting neighbouring pixels impose local constraints like smoothness but are unable to capture more complex properties such as convexity or

(a) Mean (b) MRF (c) FA

Figure 4.3: **Samples generated by widely-used models of shapes**. (a) A mean-only model. (b) A Markov random field model. (c) Discrete Factor Analysis as defined in Eqs. (4.20, 4.21).

curvature, nor can they account for longer-range properties. Carefully designed high-order potentials (e.g. Kohli et al., 2007; Komodakis and Paragios, 2009; Rother et al., 2009; Kohli et al., 2009; Nowozin and Lampert, 2009) allow particular local or longer-range shape properties to be modelled within an MRF, but these potentials fall short of capturing *all* such properties so as to make realistic-looking samples.

For example, a strong shape model of horses would know that horses have legs, heads and tails, that these parts appear in certain positions consistent with a global pose, that there are never more than four legs visible in any given image, that the legs have to support the horse's body, along with many more properties that are difficult to express in words but necessary to make the shape look plausible.

Other approaches represent shape using a parametrised contour. These have different strengths and weaknesses, but all share the fundamental challenge of imposing sufficient constraints to limit the model to valid shapes while allowing for the right degree of flexibility to capture all possible shapes. For example, a common approach when using a contour (or an image) is to use a mean shape in combination with some principal directions of variation, as captured by a Principal Components Analysis (e.g. Cootes et al., 1995; Ferrari et al., 2010) or Factor Analysis (e.g. the shape model used in FSA in Chapter 3, or the work of Cemgil et al., 2005). Such models capture the typical global shape of an object and global variations on it (such as changes in the aspect ratio of a car). However, they find it difficult to capture multimodal distributions, and tend to be poor at learning about local variations which affect only part of the shape (see e.g. incorrect generalisation in the upper-right and lower-left corners of Fig. 3.14).

Non-parametric approaches employ what is effectively a large database of template shapes (Gavrila, 2007) or shape fragments (Borenstein et al., 2004; Kumar et al., 2005).

In the former case, because no attempt is made to understand the composition of the shape, it is impossible to generalise to novel shapes not present in the database. In the latter case, the challenge lies in how to compose the shape fragments to form valid shapes. We are not aware of any method which can generate a variety of realistic looking *whole* shapes by composing fragments.

Table 4.1 and Fig. 4.3 illustrate why these existing approaches do not meet the criteria for a strong shape model.

In this chapter we consider a class of models known as *Deep Boltzmann Machines* (DBMs, Salakhutdinov and Hinton, 2009). We show how a strong model of binary shape can be constructed using a form of DBM with a set of carefully chosen capacity constraints, which we call the *Shape Boltzmann Machine* (SBM). The model is a generative model of object shape and can be learned directly from training data. The capacity constraints allow training on relatively small training sets as are common e.g. for segmentation datasets. Due to its *generative* formulation the SBM can be used flexibly, not just as a shape prior in segmentation tasks but also, for instance, to synthesise novel shapes in graphics applications, or to complete partially occluded shapes. We learn SBM models from several challenging shape datasets and evaluate them on a range of shape synthesis and completion tasks. We demonstrate that, despite the relatively small sizes of the training datasets, the learned models are both able to generate *realistic* samples and to *generalise* to generate samples that differ from images in the training dataset. We finally provide a detailed discussion of the roles played by the different capacity constraints in making the SBM work.

The remainder of this chapter is structured as follows: In Sec. 4.1 we review several families of probability distributions that have been used in the literature to model object shape. In Secs. 4.2 and 4.3 we present the SBM and describe efficient inference and learning schemes for the model. We provide an extensive experimental evaluation in Sec. 4.4, and conclude with a discussion in Sec. 4.5.

4.1 Related work

In this section we will review several undirected models suitable for modelling binary shape images. We will start with the commonly used grid-structured MRF and describe how it can be modified to form an undirected model known as the *Restricted Boltzmann*

	Realism		Generalisation
	Global	Local	
Mean *e.g. Jojic and Caspi (2004)*	✓	-	-
Deformation field *e.g. Winn and Jojic (2005)*	-	✓	✓
Factor Analysis *e.g. Cemgil et al. (2005)*	✓	-	✓
Fragments *e.g. Borenstein et al. (2004)*	-	✓	✓
Grid MRFs/CRFs *e.g. Rother et al. (2004)*	-	✓	✓
High-order potentials *e.g. Nowozin and Lampert (2009)*	limited	✓	✓
Database *e.g. Gavrila (2007)*	✓	✓	-
Shape Boltzmann Machine	✓	✓	✓

Table 4.1: **Comparison of a number of different shape models**.

Machine (RBM). We then describe how RBMs can be stacked to form the hierarchical structure of the *Deep Boltzmann Machine* (DBM).

We will specify undirected models in terms of an energy function $E(x_1, \ldots, x_N)$ defined over the relevant set of random variables $x_1, \ldots, x_N$ (image pixels, possibly latent variables). The associated Gibbs distribution is then given by

$$p(x_1, \ldots, x_N) = \frac{1}{Z} \exp -E(x_1, \ldots, x_N), \tag{4.1}$$

where

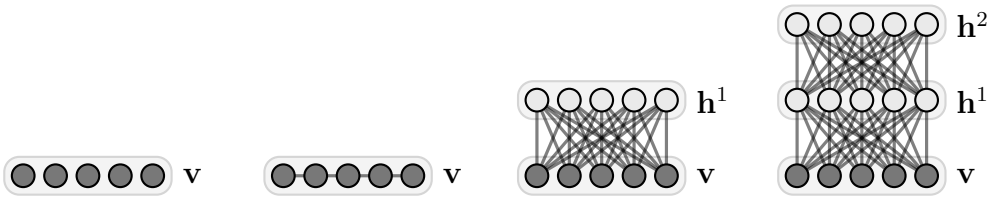
$$Z = \sum_{x_1, \ldots, x_N} \exp -E(x_1, \ldots, x_N) \tag{4.2}$$

is the normalisation constant. We will further use v_i to denote image pixel i, and $\mathbf{v} = (v_i)^T$ to denote a column-vector of image pixels[1]. The pixels are assumed to be binary, and we consider categorical pixels in Sec. 5.1.1. Similarly we use h_j and $\mathbf{h} = (h_j)^T$ to refer to binary hidden variable j and a vector of hidden variables respectively.

4.1.1 Grid Markov random fields

The simplest approach is to model each shape pixel v_i independently with categorical variables whose parameters are specified by the object's mean shape (Fig. 4.4a). Such

[1] In Chapter 3 we used $\mathbf{s}_i$ to denote the discrete variable that assigns RGB pixel i to a part. This is analogous to the v_i here. We use v_i instead of $\mathbf{s}_i$ as this notation is more commonly used in the deep learning literature.



(a) Mean (b) MRF (c) RBM (d) DBM

Figure 4.4: **Undirected models of shape**. Dark circles represent image pixels and light circles represent hidden variables. (a) 1D slice of a mean model. (b) Markov random field in 1D. (c) Restricted Boltzmann Machine in 1D. (d) Deep Boltzmann Machine in 1D.

a 'mean model' can be expressed in terms of an energy function comprised of single-variable terms only:

$$E(\mathbf{v}|\boldsymbol{\theta}) = \sum_i f_i(v_i|b_i). \tag{4.3}$$

For binary images, for instance, the f_i might take the form $f_i(v_i|b_i) = -b_i v_i$, specifying the unnormalised log-probability of $v_i = 1$ which results in the normalised probability being $p(v_i = 1|b_i) = \exp(b_i)/(1 + \exp(b_i))$.

A binary grid-structured MRF defines a distribution over binary images $\mathbf{v}$ whose energy function is

$$E(\mathbf{v}|\boldsymbol{\theta}) = \sum_i f_i(v_i|b_i) + \sum_{(i,j)} f_{ij}(v_i, v_j|w_{ij}), \tag{4.4}$$

where i ranges over image pixels, (i, j) ranges over grid edges between pixels i and j and the potentials are parametrised by b_i and w_{ij}, again jointly denoted by $\boldsymbol{\theta}$. The grid structure of the MRF arises from the pairwise potentials f_{ij} shown in Fig. 4.4b. These potentials induce dependencies between neighbouring pixels that can favour local shape properties such as connectedness or smoothness, but it is commonly accepted that grid-structured, pairwise MRFs are limited models of global shape (e.g. Morris et al., 1996; Tjelmeland and Besag, 1998).

In an attempt to capture more complex or global shape properties, much recent research has therefore focused on constructing higher-order potentials (HOPs), which take the configuration of larger groups of image pixels into account (i.e. their energy includes potentials f that depend on more than two pixel variables). The maximum number of variables per potential is referred to as the 'order' of the model. Since, in general, the cost of naïve inference (e.g. finding the most likely (MAP) configuration of the

variables) in MRFs grows exponentially in the model order, there has been a strong emphasis on developing higher-order potentials for which efficient inference schemes can be devised.

The higher order potentials in Rother et al. (2009), for instance, are defined in terms of a set of 'reference patterns' and penalise deviations of groups of pixels from these patterns. Such HOPs can be considered to be introducing an auxiliary hidden variable connected through pairwise potentials to multiple image pixels (Li et al., 2013). The introduction of such hidden variables provides a powerful way to capture and learn complex properties of multiple image pixels. When such hidden variables are marginalised out they induce high-order constraints amongst the image pixels. Yet, because the model only contains pairwise potentials, both learning and inference remain tractable.

4.1.2 Restricted Boltzmann Machines

One model that makes heavy use of hidden variables to introduce dependencies between the observed variables is the *Restricted Boltzmann Machine* (RBM, e.g. Freund and Haussler, 1994). In an RBM, a number of hidden variables $\mathbf{h}$ are used, each of which is connected to all image pixels as shown in Fig. 4.4c. However, unlike a grid MRF, there are no direct connections between the image pixels $\mathbf{v}$. There are also no direct connections between the hidden variables. Hence, the energy function takes the form:

$$E(\mathbf{v}, \mathbf{h}|\boldsymbol{\theta}) = \sum_i b_i v_i + \sum_{i,j} w_{ij} v_i h_j + \sum_j c_j h_j, \tag{4.5}$$

where i now ranges over pixels and j ranges over hidden variables. The key points to note are that the potential functions are all simple products and that the only pairwise potentials are those between each visible and each hidden variable. By learning the parameters of the potentials $\{w_{ij}, b_i, c_j\}$, the model can learn about high-order constraints in the data set.

The effect of the latent variables can be directly appreciated by considering the marginal distribution over $\mathbf{v}$ which is given by marginalising over the hidden variables:

$$p(\mathbf{v}|\boldsymbol{\theta}) = \sum_{\mathbf{h}} \frac{1}{Z(\boldsymbol{\theta})} \exp\{-E(\mathbf{v}, \mathbf{h}|\boldsymbol{\theta})\}, \tag{4.6}$$

where the normalisation constant $Z(\boldsymbol{\theta})$ is given by

$$Z(\boldsymbol{\theta}) = \sum_{\mathbf{v},\mathbf{h}} \exp\{-E(\mathbf{v}, \mathbf{h}|\boldsymbol{\theta})\}. \tag{4.7}$$

This marginalisation allows the model to capture high-order dependencies between the visible units. In fact, the hidden units can be summed out analytically (e.g. Freund and Haussler, 1994), giving rise to an alternative formulation of the RBM in terms of high-order potentials that no longer includes latent variables. The energy of this marginal distribution is given by

$$E(\mathbf{v}|\boldsymbol{\theta}) = \sum_i f_i(v_i; b_i) + \sum_j g_j(\mathbf{v}|W_{\cdot j}), \tag{4.8}$$

where

$$f_i(v_i|b_i) = -b_i v_i \tag{4.9}$$

and

$$g_j(\mathbf{v}) = -\log(1 + \exp(\sum_i w_{ij} v_i + c_j)). \tag{4.10}$$

It is instructive to compare the form of Eq. 4.8 with the energy of the grid-structured MRF in Eq. 4.4: Whereas the energy of the grid-structured MRF was comprised of unary and pair-wise terms only ($f_i(v_i)$ and $f_{ij}(v_i, v_j)$ respectively), the energy of the RBM involves unary potentials *as well as high-order* potentials, each of which is defined over all pixels $\mathbf{v}$ (the $g_j(\mathbf{v})$). There is one such high-order potential for each hidden unit, and it is these high-order potentials that allow the RBM to model considerably more complicated dependencies than, for instance, pairwise MRFs.

Whilst marginalisation over the latent variables makes the high-order potentials explicit, the formulation that includes latent variables suggests an efficient inference scheme (in loose analogy to the use of latent variables for the HOPs discussed in Sec. 4.1.1): When written as in Eq. 4.5 the RBM forms a bipartite graph that has edges only between hidden and visible variables. As a consequence all hidden units are conditionally independent given the visible units – and vice versa. This property can be exploited to make inference exact and efficient. The conditional probabilities are

$$p(v_i = 1|\mathbf{h}) = \sigma(\sum_j w_{ij} h_j + b_i), \tag{4.11}$$

$$p(h_j = 1|\mathbf{v}) = \sigma(\sum_i w_{ij} v_i + c_j), \tag{4.12}$$

where $\sigma(y) = 1/(1 + \exp(-y))$ is the sigmoid function. This property allows for efficient implementations of block-Gibbs sampling where all $\mathbf{v}$ and all $\mathbf{h}$ are sampled in parallel in an alternating manner, which can be exploited during approximate learning (Hinton, 2002; Tieleman, 2008).

4.1.3 Deep Boltzmann Machines

RBMs can, in principle, approximate any binary distribution (Freund and Haussler, 1994; Le Roux and Bengio, 2008), but this can require an exponential number of hidden units and a similarly large amount of training data. The DBM provides a richer model by introducing additional layers of latent variables as shown in Fig. 4.4d. The additional layers capture high-order dependencies between the hidden variables of previous layers and so can learn about complex structure in the data using relatively few hidden units. The energy of a DBM with two layers of latent variables is given by

$$\begin{aligned} E(\mathbf{v}, \mathbf{h}^1, \mathbf{h}^2) &= \sum_i b_i v_i + \sum_{i,j} w_{ij}^1 v_i h_j^1 + \sum_j c_j^1 h_j^1 \\ &+ \sum_{j,k} w_{jk}^2 h_j^1 h_k^2 + \sum_k c_k^2 h_k^2. \end{aligned} \tag{4.13}$$

As for the RBM, the posterior distribution over the visibles is obtained by marginalisation, this time with respect to both sets of hidden variables:

$$p(\mathbf{v}|\boldsymbol{\theta}) = \sum_{\mathbf{h}^1, \mathbf{h}^2} \frac{1}{Z(\boldsymbol{\theta})} \exp\{-E(\mathbf{v}, \mathbf{h}^1, \mathbf{h}^2|\boldsymbol{\theta})\}, \tag{4.14}$$

and the normalisation constant defined analogously:

$$Z(\boldsymbol{\theta}) = \sum_{\mathbf{v}, \mathbf{h}^1, \mathbf{h}^2} \exp\{-E(\mathbf{v}, \mathbf{h}^1, \mathbf{h}^2|\boldsymbol{\theta})\}. \tag{4.15}$$

Although exact inference is no longer possible in this model, the conditional distributions $p(\mathbf{v}|\mathbf{h}^1)$, $p(\mathbf{h}^1|\mathbf{v}, \mathbf{h}^2)$, and $p(\mathbf{h}^2|\mathbf{h}^1)$ remain factorised due to the layering:

$$p(v_i = 1|\mathbf{h}^1) = \sigma(\sum_j w_{ij}^1 h_j^1 + b_i), \tag{4.16}$$

$$p(h_j^1 = 1|\mathbf{v}, \mathbf{h}^2) = \sigma(\sum_i w_{ij}^1 v_i + \sum_k w_{jk}^2 h_k^2 + c_j^1), \tag{4.17}$$

$$p(h_k^2 = 1|\mathbf{h}^1) = \sigma(\sum_j w_{jk}^2 h_j^1 + c_k^2). \tag{4.18}$$

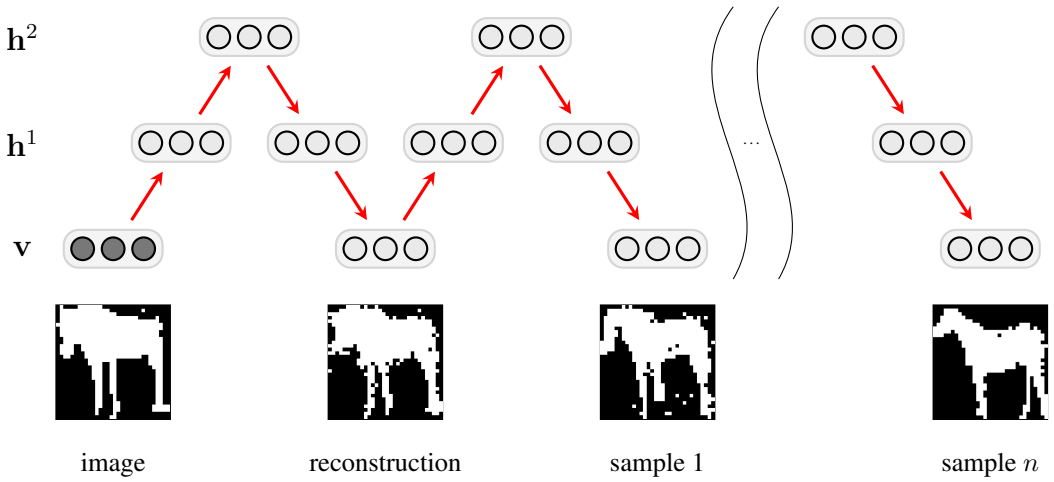


Figure 4.5: **DBM Markov Chain Monte-Carlo**. Block-Gibbs MCMC sampling scheme, in which $\mathbf{v}$, $\mathbf{h}^1$ and $\mathbf{h}^2$ variables are sampled in turn. Note that each sample of $\mathbf{h}^1$ is obtained conditioned on the current state of $\mathbf{v}$ *and* $\mathbf{h}^2$. For sufficiently large values of n, sample n will be uncorrelated with the original image.

This allows for computationally efficient inference, either by layerwise block-Gibbs sampling from the posterior $p(\mathbf{h}^1, \mathbf{h}^2|\mathbf{v})$ (Fig. 4.5), or by using a mean field procedure with a fully factorised approximate posterior as described in Salakhutdinov and Hinton (2009). The layering further admits a layer-wise pre-training procedure that makes it less likely that learning will get stuck in local optima. Hence the DBM is both a rich model of binary images and a tractable one.

4.2 Model

RBMs and DBMs are powerful generative models, but also have many parameters. Since they are typically trained on large amounts of unlabelled data (thousands or tens of thousands of examples), this is usually less of a problem than in supervised settings. Segmented images, however, are expensive to obtain and datasets are typically small (hundreds of examples). In such a regime, RBMs and DBMs can be prone to overfitting, making them an unusual choice for applications such as ours.

In this section we will describe how we can impose a set of carefully chosen connectivity and capacity constraints on a DBM to overcome this problem. The resulting SBM formulation not only learns a model that accurately captures the properties of binary

shapes, but that also generalises well, even when trained on small datasets.

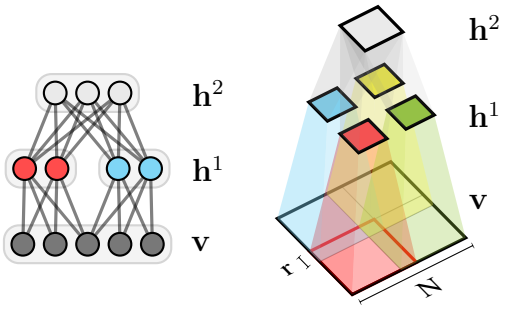
4.2.1 The Shape Boltzmann Machine

The SBM used here has two layers of latent variables: $\mathbf{h}^1$ and $\mathbf{h}^2$. The visible units $\mathbf{v}$ are the pixels of a binary image of size $N \times M$. In the first layer we enforce local receptive fields by connecting each hidden unit in $\mathbf{h}^1$ only to a subset of the visible units, corresponding to one of four rectangular patches, as shown in Fig. 4.6. In order to encourage boundary consistency each patch overlaps its neighbour by r pixels and so has side lengths of $N/2 + r/2$ and $M/2 + r/2$. We furthermore share weights between the four sets of hidden units and patches, however the visible biases b_i are not shared.

Similar constraints have previously been used in the literature (e.g. Desjardins and Bengio, 2008; Raina et al., 2009; Lee et al., 2009; Norouzi et al., 2009; Ranzato et al., 2010, 2011), especially in convolutional and tiled-convolutional formulations of RBMs and DBNs. In comparison, in the SBM the receptive field overlap of adjacent groups of hidden units is particularly small compared to their sizes.

Overall, these modifications reduce the number of first layer parameters by a factor of about 16 which reduces the amount of data needed for training by a similar factor. At the same time these modifications take into account two important properties of shapes: First, the restricted receptive field size reflects the fact that the strongest dependencies between pixels are typically local, while distant parts of an object often vary more independently (the small overlap allows boundary continuity to be learned primarily at the lowest layer); second, weight sharing takes account of the fact that many generic properties of shapes (e.g. smoothness) can potentially be independent of the image position.

For the second layer we choose full connectivity between $\mathbf{h}^1$ and $\mathbf{h}^2$, but restrict the relative capacity of $\mathbf{h}^2$. We use around 4×500 hidden units for $\mathbf{h}^1$ vs. around 50 for $\mathbf{h}^2$ in our single class experiments. While the first layer is primarily concerned with generic, local properties, the role of the second layer is to impose global constraints, e.g. with respect to the class of an object shape or its overall pose. The second layer mediates dependencies between pixels that are far apart (not in the same local receptive field), but these dependencies will be weaker than between nearby pixels that share



(a) (b)

Figure 4.6: **The Shape Boltzmann Machine** . (a) 1D slice of a Shape Boltzmann Machine. (b) The Shape Boltzmann Machine in 2D.

first-level hidden units. Limiting the capacity of the second-layer encourages this division of labour and helps to prevent the model from overfitting to small training sets. Note that this is in contrast to Salakhutdinov and Hinton (2009) who use a top-most layer that is at least as large as all of the preceding layers.

4.3 Learning

Learning of the model involves maximising $\log p(\mathbf{v}|\boldsymbol{\theta})$ of the observed data $\mathbf{v}$ with respect to its parameters $\boldsymbol{\theta} = \{\mathbf{b}, W^1, W^2, \mathbf{c}^1, \mathbf{c}^2\}$ (see Eqs. 4.6, 4.14). The gradient of the log-likelihood of a single training image with respect to the parameters is given by

$$\begin{aligned} \nabla_{\boldsymbol{\theta}} \log p(\mathbf{v}|\boldsymbol{\theta}) &= \langle \nabla_{\boldsymbol{\theta}} E(\mathbf{v}', \mathbf{h}^1, \mathbf{h}^2|\boldsymbol{\theta}) \rangle_{p_{\boldsymbol{\theta}}(\mathbf{v}', \mathbf{h}^1, \mathbf{h}^2)} \\ &- \langle \nabla_{\boldsymbol{\theta}} E(\mathbf{v}, \mathbf{h}^1, \mathbf{h}^2|\boldsymbol{\theta}) \rangle_{p_{\boldsymbol{\theta}}(\mathbf{h}^1, \mathbf{h}^2|\mathbf{v})}, \end{aligned} \tag{4.19}$$

and the total gradient is obtained by summing the gradients of the individual training images (e.g. Ackley et al., 1985; Freund and Haussler, 1994; Salakhutdinov and Hinton, 2009). The first term on the right hand side is the expectation of the gradient of the energy (see Eq. 4.13) where the expectation is taken with respect to the joint distribution over $\mathbf{v}$, $\mathbf{h}^1$, $\mathbf{h}^2$ defined by the model. The second term is also an expectation of the gradient of the energy, but this time taken with respect to the posterior distribution over $\mathbf{h}^1$, $\mathbf{h}^2$ given the observed image $\mathbf{v}$.

Although the gradient is readily written out, maximisation of the log-likelihood is difficult in practice for three reasons: 1) except for very simple cases the gradient is

intractable to compute as both expectations involve a sum over a number of terms that is exponential in the number of variables (visible and hidden units); 2) the mere presence of latent variables; and 3) gradient ascent in the likelihood is prone to getting stuck in local optima.

For the SBM we closely follow the procedure proposed in Salakhutdinov and Hinton (2009) which minimises these difficulties in three ways: 1) it approximates the first expectation in Eq. 4.19 with samples drawn from the model distribution via MCMC; 2) it approximates the second expectation using a mean-field approximation to the posterior; and 3) it employs a pre-training strategy that provides a good initialisation to the weights W^1, W^2 before attempting learning in the full model.

Learning proceeds in two phases. In the *pre-training* phase we greedily train the model bottom up, one layer at a time. The purpose of this phase is to find good initial values for all parameters of the model. We begin by training an RBM on the observed data.

The likelihood gradient of an RBM takes a form similar to Eq. 4.19. Unlike for the DBM, for an RBM the second expectation over the conditional distribution of the hidden units $\mathbf{h}$ given the data is tractable and can be computed exactly (see Eq. 4.12).

The first expectation, taken with respect to the full model distribution, however, remains intractable. We therefore perform stochastic maximum likelihood learning (SML, also referred to as 'persistent contrastive divergence'; Neal, 1992; Tieleman, 2008; Salakhutdinov and Hinton, 2009) where this expectation is approximated using samples from the model distribution obtained via MCMC. While a naïve MCMC approximation of the expectation would be computationally expensive, considerable computational savings can be obtained through a set of Markov chains that are initialised at the beginning of learning and then maintained over the course of learning (hence the adjunct 'persistent'), alternating updates of the model parameters $\boldsymbol{\theta}$ with Gibbs sampling steps to update the sample approximation to the model distribution. This algorithm is an instance of a stochastic approximation scheme of the Robbins-Monro type (Robbins and Monro, 1951; Younes and Sud, 1989; Younes, 1999).

The number of hidden units of this RBM is the same as the size of $\mathbf{h}^1$ in the full SBM model and it obeys the same connectivity constraints as the SBM's first layer. Once this RBM is trained, we infer the conditional mean of the hidden units using Eq. 4.12 for each training image. The resulting vectors then serve as the training data for a second RBM with the same number of hidden units as $\mathbf{h}^2$, which is trained using SML.

We use the parameters of these two RBMs to initialise the parameters of the full SBM model as described in Salakhutdinov and Hinton (2009). Simply speaking, we use the weights of the first RBM to initialise the parameters of the lower layer of the SBM ($\mathbf{b}$ and W^1), and the parameters of the second RBM to initialise the upper layer (W^2 and $\mathbf{c}^2$). As discussed in detail in Salakhutdinov and Hinton (2009) special care must be taken to account for the fact that in the full model $\mathbf{h}^1$ now receives input from both $\mathbf{v}$ and $\mathbf{h}^2$.

In the second phase we then perform approximate stochastic gradient ascent in the likelihood of the full model to fine-tune the parameters in an expectation-maximisation-like scheme. This involves the same sample-based approximation to the gradient of the normalisation constant used for learning the RBMs (Tieleman, 2008; Salakhutdinov and Hinton, 2009), as well as a fully factorised mean-field approximation to the posterior $p(\mathbf{h}^1, \mathbf{h}^2|\mathbf{v})$. This joint training is essential to separate out learning of local and global shape properties into the two hidden layers.

4.4 Experiments

We perform an extensive experimental evaluation of the SBM model on five datasets in total. The presentation of the results is divided into three parts:

In Sec. 4.4.1 we focus on demonstrating that the SBM can indeed act as a strong model of object shape. For this purpose we perform qualitative and quantitative evaluations on two challenging datasets: the Weizmann horse datasets and motorbikes from Caltech-101. Despite both datasets being relatively small we find that the learned models capture essential high- and low-level properties of the shapes in the training data, producing realistic samples and generalising to novel shapes not present in the training data. Quantitatively we find that the SBM outperforms several baseline models in a difficult shape completion task.

The goal of Sec. 4.4.2 is to examine the contribution of our architectural choices to the success of the SBM. We address the impact of localised receptive fields, weight-sharing, and of the hierarchical structure of the model.

In many situations it is desirable or even necessary to model not just a single but multiple object classes with the same model. In Sec. 4.4.3 we therefore introduce an additional dataset comprised of multiple object categories (Weizmann horses and several

animals from Caltech-101) and demonstrate that the SBM, with a single set of parameters, can learn a joint model of several categories from unlabelled data, generalising reliably within each category.

4.4.1 Generalisation and Realism

In this section we demonstrate that the SBM can be trained to be a strong model of object shape. For this purpose we consider two challenging datasets: Weizmann horses and Caltech-101 motorbikes.

4.4.1.1 Weizmann horse dataset

The Weizmann horse dataset (Borenstein et al., 2004) contains 327 images, all of horses facing to the left, but in a variety of poses. The dataset is challenging because in addition to their overall pose variation, the positions of the horses' heads, tails and legs change considerably from image to image. Compared to the amount of variability seen in the data, the number of training images is small.

The binary images are cropped and normalised to 32×32 pixels (see Fig. 4.7a). We trained an SBM with overlap $r = 4$, and 2,000 and 100 units for $\mathbf{h}^1$ and $\mathbf{h}^2$ respectively. The first layer was pre-trained for 3,000 epochs (iterations) and the second layer for 1,000 epochs. After pre-training, joint training was performed for 1,000 epochs. Our MATLAB implementation completed training in around 4 hours, running on a dual-core, 3GHz PC with 4GB of memory.

4.4.1.2 Caltech motorbikes dataset

Our second dataset is based on Caltech-101 (Fei-Fei et al., 2004), and consists of 798 motorbike silhouettes. These binary images are of higher resolution than the horses and are cropped and normalised to 64×64 pixels (see Fig. 4.9a). We trained an SBM with overlap $r = 4$, and 1200 and 50 units for $\mathbf{h}^1$ and $\mathbf{h}^2$ respectively, using the same schedule as before.

It is noteworthy that for both datasets the number of training images is relatively small compared to the variability present in the data and, in particular, compared to the size of datasets that deep learning models are typically trained on. Both datasets consist of

significantly less than 1,000 training images which is in stark contrast to the several thousand or, more often, tens of thousands of training images for most applications of deep models in the literature. Salakhutdinov and Hinton (2009), for instance, use the 60,000 training images from the MNIST dataset for their experiments.

4.4.1.3 Baseline models

For comparison we considered two baseline models. First, we trained a Factor Analysis (FA) model with 10 latent dimensions. The FA model was modified to work on discrete binary images by linearly mixing the independent Gaussian latent variables and then passing them through a sigmoid to obtain binary observed variables (similar to Clipped Factor Analysis, Cemgil et al., 2005, and the shape component of the FSA model described in Sec. 3.1.1):

$$p(\mathbf{h}) = \mathcal{N}(\mathbf{0}, \mathbf{I}) \tag{4.20}$$

$$p(v_i = 1|\mathbf{h}) = \sigma(\sum_j w_{ij} h_j + b_j), \tag{4.21}$$

where $\mathbf{0}$ is a vector of zeros and $\mathbf{I}$ denotes the identity matrix. The model was trained using gradient ascent, and inference was performed using elliptical slice sampling as described in Sec. 3.2.

Our second baseline model was the RBM as defined in Eq. 4.5. We used 500 hidden units and trained the model using SML as described in Sec. 4.3. For both baseline models the hyperparameters and number of hidden units were manually optimised for the visual quality of their samples for each dataset.

4.4.1.4 Realism

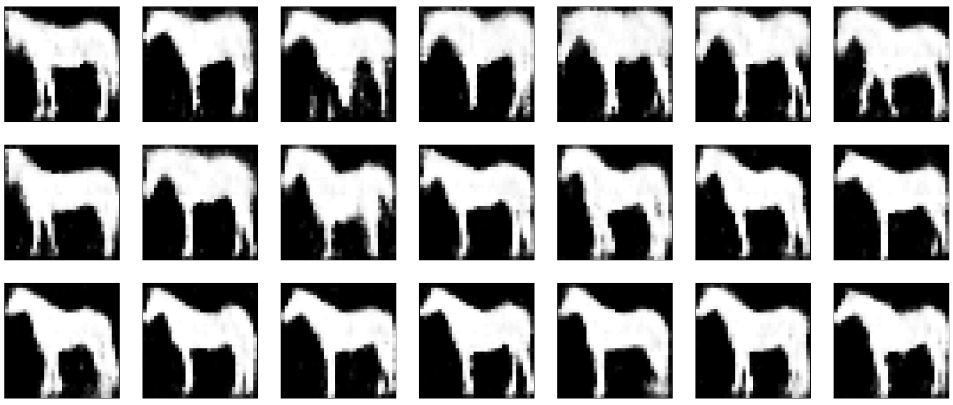
To assess the Realism requirement, we sampled a set of shapes from each model, as shown in Fig. 4.7 and Fig. 4.9 for the horse and motorbike datasets respectively.

The FA shape models can be sampled from directly. For the RBM and SBM models samples are generated by extended block Gibbs sampling. In particular, for the SBM models samples were generated using the scheme outlined in Fig. 4.5. As is common in the literature, we visualise the samples by showing for each pixel i the (grayscale) *conditional probability* of that pixel $p(v_i = 1|\mathbf{h})$ given the particular hidden configuration that constitutes the current state of the Markov chain. Binary samples can be

(a) Data

(b) Factor Analysis

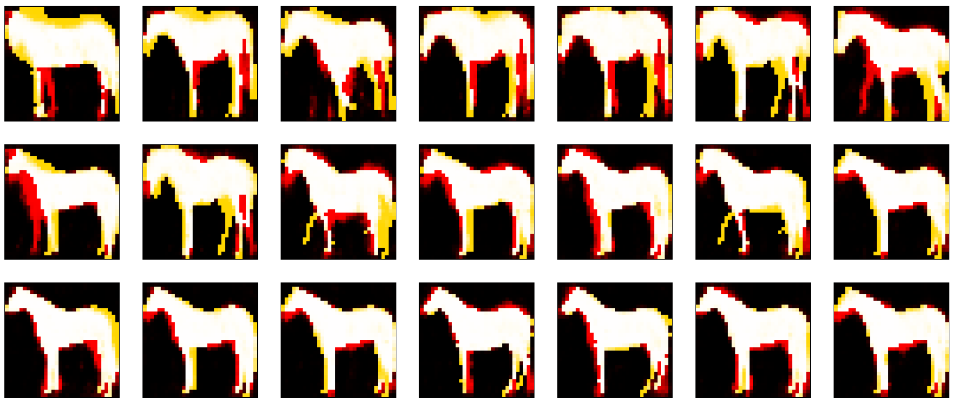
(c) Restricted Boltzmann Machine



(d) Shape Boltzmann Machine

Figure 4.7: **Sampled horses**. (a) A selection of images from the Weizmann horse dataset. (b) A collection of samples from a discrete Factor Analysis model. The Gaussianity assumption forces the model to allocate probability mass to unlikely horse shapes. (c) Samples from an RBM. (d) Samples from an SBM. The model generates samples of varying pose, with the correct numbers of legs and details are preserved (samples are arranged left-right, up-down in decreasing order of generalisation).

(a) Sample - Closest - Generalisation



(b) Generalisations

Figure 4.8: **Generalisation**. (a) A sample from the SBM, the closest image in the training dataset to the generated sample, and the difference between the two images. Red pixels have been generated by the sample but are absent in the training image; yellow pixels are present in the training image but absent in the sample. The model has generalised to an unseen, but realistic horse shape. (b) Generalisations made in each of the samples in Fig. 4.7d.

generated per-pixel from a Bernoulli distribution where the gray level specifies the distribution mean.

FA effectively defines a transformed Gaussian distribution over the image pixels and is thus inherently unimodal. In order to account for the diversity of shapes in the training data it is therefore forced to allocate probability mass to images that do not correspond to realistic horse or motorbike shapes, as shown in Figs. 4.7b and 4.9b.

By contrast, the RBM can, in principle, account for multi-modal data and could thus assign probability mass more selectively. However, as the samples of horses (Fig. 4.7c) indicate, the model also fails to learn a good model of the variability of horse shapes – the samples are mostly of the same pose, and details of the shape are lost when the pose changes. We found this effect to be even more dramatic for RBM samples of motorbikes, due to the larger image size (see Fig. 4.9c).

These problems are symptomatic of training RBMs with insufficient data. The SBM aims to overcome these problems through a combination of connectivity constraints, weight sharing, and model hierarchy. As we will discuss in more detail in Sec. 4.4.2, the combination of these ingredients is necessary to obtain a strong model of shape.

Increasing the number of hidden units in the hope of learning more local filters did not solve the problem, confirming that the lack of data is the issue. An RBM with similar connectivity constraints as the first layer of the ShapeBM has fewer parameters than a fully connected RBM and thus suffers less from overfitting, but without the second layer it fails to account for global constraints on the shape.

Samples from the SBM for horses and motorbikes are shown in Figs. 4.7d and 4.9d respectively. First, we note that the model generates natural shapes from a variety of poses. Second, we observe that details such as legs (in the case of horses) or handle bars, side mirrors, and forks (in the case of motorbikes) are preserved and remain sharply defined in the samples. Third, we note that the horses have the correct number of legs while motorbikes have, for instance, the correct number of handle bars and wheels. Finally, we note that the patch overlap ensures seamless connections between the four quadrants of the image. Indeed, horse and motorbike samples generated by the model look sufficiently realistic that we consider the model to have fulfilled the Realism requirement.

4.4.1.5 Generalisation

We next investigated to what extent the SBM meets the Generalisation requirement, to ensure that the model has not simply memorised the training data. In Fig. 4.8 we show for horses the difference between the sampled shapes from Fig. 4.7d and their closest images in the training set. We use the Hamming distance between training images and a thresholded version of the conditional probability (> 0.3), as the similarity measure. This measure was found to retrieve the visually most similar images.

Red indicates pixels that are in the sample but not in the closest training image, and yellow indicates pixels in the training image but not in the sample. Fig. 4.9e shows a similar analysis for samples from the model learned for motorbikes. Both models generalise from the training data-points in non-trivial ways whilst maintaining validity of the overall object shape. These results suggest that the SBM generalises to realistic shapes that it has not encountered in the training set.

4.4.1.6 Shape completion

We further assessed both the realism and generalisation capabilities of the SBM by using it to perform shape completion, where the goal is to generate likely configurations of pixels for a missing region of the shape, given the rest of the shape. To perform completion we obtain samples of the missing – or unobserved – pixels $\mathbf{v}_U$ conditioned on the remaining (observed) pixels $\mathbf{v}_O$ (U and O denote the set indices of unobserved and observed pixels respectively). This is achieved using a Gibbs sampling procedure that samples from the conditional distribution. In this procedure, samples are obtained by running a Markov chain as before, sampling $\mathbf{v}$, $\mathbf{h}^1$, and $\mathbf{h}^2$ from their respective conditional distributions, but every time $\mathbf{v}$ is sampled we 'clamp' the observed pixels $\mathbf{v}_O$ of the image to their given values, updating only the state of the unobserved pixels $\mathbf{v}_U$. Since the model specifies a *distribution* over the missing region $p(\mathbf{v}_U|\mathbf{v}_O)$, multiple such samples capture the variability of possible solutions that exist for any given completion task. In Fig. 4.10 we show how the samples become more constrained as the missing region shrinks. Fig. 4.11 and Fig. 4.12 show sampled completions of regions of horse and motorbike images that the model had not seen during training. Despite the large sizes of the missing portions, and the varying poses of the horses and motorbikes, completions look realistic.

(a) Training

(b) Factor Analysis

(c) Restricted Boltzmann Machine

(d) Shape Boltzmann Machine

(e) Shape Boltzmann Machine differences

Figure 4.9: **Results on Caltech-101 motorbikes**. (a) A selection of images from the training set (at 64×64 pixels). (b) A set of samples from the FA baseline model. (c) A set of samples from the RBM baseline model. (d) A chain of samples generated by the SBM. (e) Difference images for each of the samples in (d) (same format as in Fig. 4.8). The model generalises from training examples in non-trivial ways, whilst maintaining overall motorbike look-and-feel.

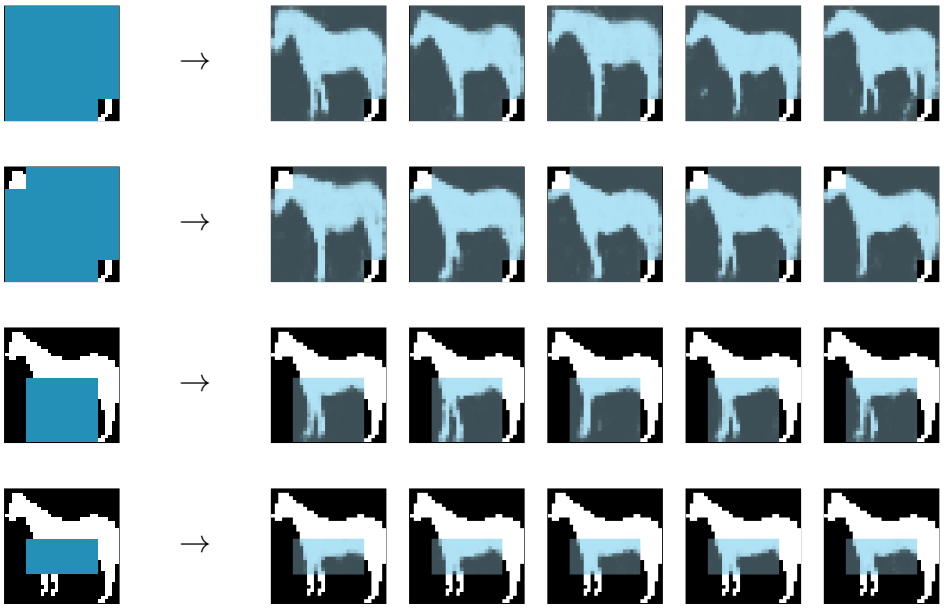


Figure 4.10: **Shape completion variability**. Blue in the first column indicates the missing regions. The samples highlight the *variability* in possible completions captured by the model. As the missing region shrinks, the samples become more constrained.

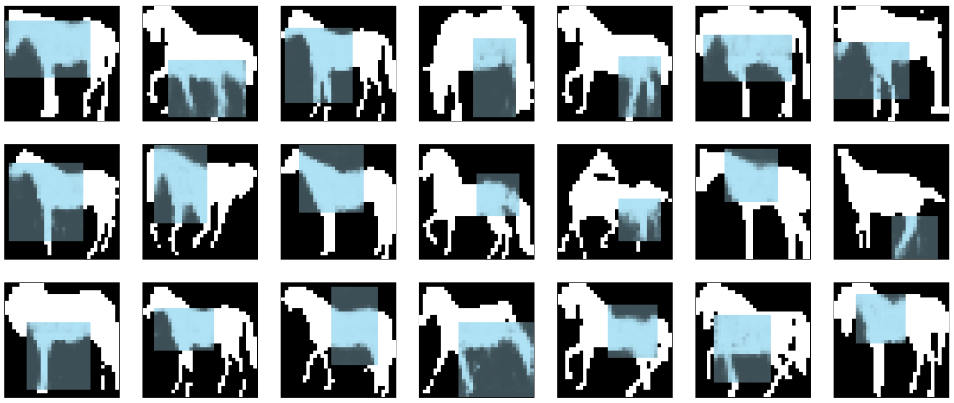


Figure 4.11: **Sampled image completion for horses**. The SBM completes rectangular imputations of random size on images not seen during training.

Figure 4.12: **Sampled image completion for motorbikes**. The SBM completes rectangular imputations of random size on images not seen during training.

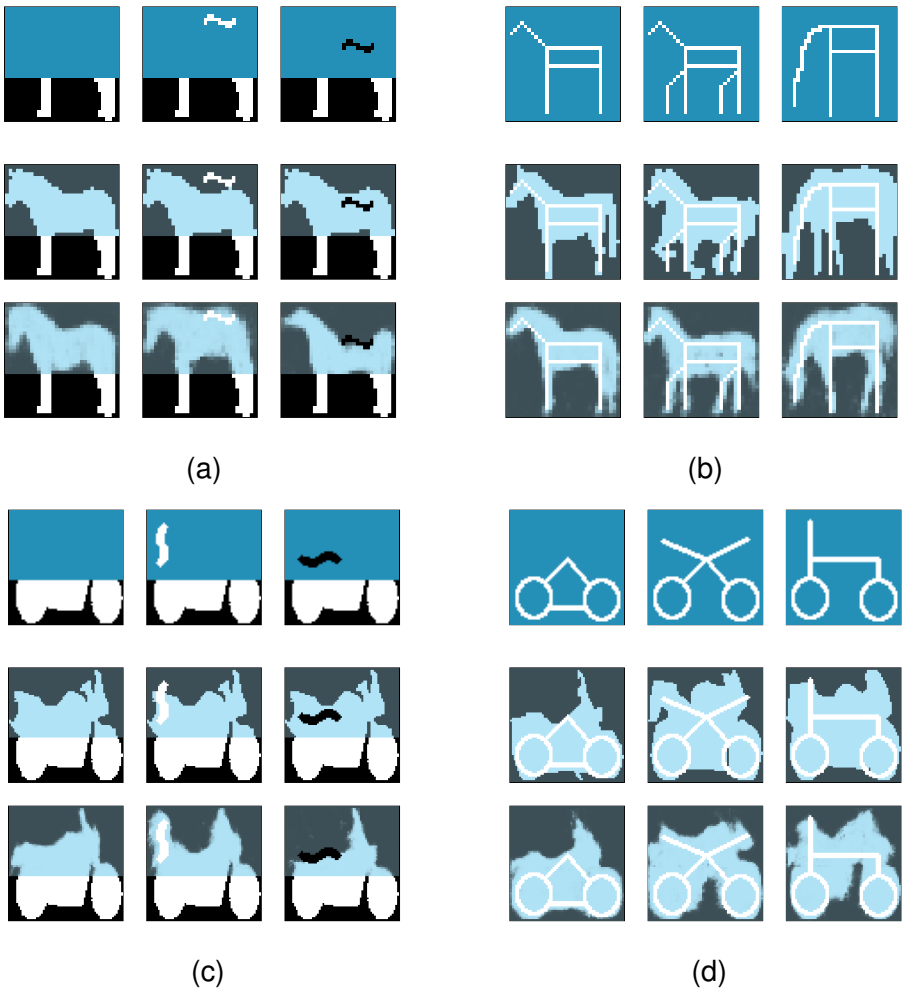


Figure 4.13: **Constrained shape completion**. Missing regions (blue pixels, top row) are completed by finding the closest match to the prescribed pixels in the training data (middle row) and using the SBM (bottom row). (a) The horse's back is pulled up by the SBM (bottom row) using an appropriate 'on' brush. Notice how the stomach moves up and the head angle changes to maintain a valid shape. The horse's back is then pushed down with an 'off' brush. (b) Given only minimal user input, the model completes the images to generate realistic horse shapes. (c,d) Motorbikes (at 64×64). In many cases, the nearest neighbour method fails to find a suitable training image to satisfy the constraints.

The SBM's ability to do shape completion suggests applications in a computer graphics setting. Sampled completions can be constrained in real-time by simply clamping certain pixels of the image. In Fig. 4.13a and Fig. 4.13c we show snapshots of a graphical user interface in which the user modifies a horse or motorbike silhouette with a digital brush. The model's ability to generalise enables it to generate samples that satisfy the user's constraints. The model's accurate knowledge about horse and motorbike shapes ensures that the samples remain realistic.

As a direct comparison we also consider a simple data-base driven ('non-parametric') approach where we try to find suitable completions via a nearest-neighbour search in our database of training shapes. As shown in Fig. 4.13 such a database-driven approach can fail to find shapes that match the constraints.

The same approach can also be used to generate complete silhouettes in different poses given simple stick figures provided by the user (see Figs. 4.13b and 4.13d). This GUI and a video showing its use may be downloaded from `http://bit.ly/ShapeBM`.

4.4.1.7 Quantitative comparison

A natural way to directly evaluate a generative model *quantitatively* is by computing the likelihood of some held-out data under the model. Unfortunately, this likelihood computation is intractable for DBMs. Approximations, e.g. based on annealed importance sampling, (Neal, 2001; Salakhutdinov and Murray, 2008; Salakhutdinov and Hinton, 2009; Murray and Salakhutdinov, 2009) are computationally very expensive and their accuracy can be difficult to assess.

As an alternative we therefore introduce what we will refer to as an 'imputation score' for the shape completion task as a measure of the strength of a model. We collect additional horse and motorbike silhouettes from the web (25 horses and 25 motorbikes), and divide each into 9 segments. We then perform multiple imputation tests for each image. In each test, we remove one of the segments and estimate the conditional probability of that segment under the model, given the remaining 8 segments. The log probabilities are then averaged across the different segments and images to give the score.

Except for the mean model (where they are trivial) the conditional distributions over the subsets of unobserved pixels given the rest of the image are infeasible to compute

in practice due to the dependencies introduced by the latent variables. We therefore approximate the required conditional log-probabilities via MCMC: For a particular image and segment we draw configurations of the latent variables from the posterior given the observed part of the image and then evaluate the conditional probability of the true configuration of the unobserved segment given the latent variables, i.e. we compute

$$p(\mathbf{v}_U|\mathbf{v}_O) \approx \frac{1}{S}\sum_s p(\mathbf{v}_U|\hat{\mathbf{h}}^s), \tag{4.22}$$

where $\mathbf{v}_U$ and $\mathbf{v}_O$ indicate the set of unobserved/observed pixels (corresponding to the one removed and the 8 remaining segments), and $\hat{\mathbf{h}}^s \sim \mathbf{h}|\mathbf{v}_O$ are samples from the conditional distribution over the hidden units given the observed part of the image obtained via MCMC[2]. Provided that our MCMC scheme allows us to sample from the true posterior the right hand side of Eq. 4.22 provides us with an unbiased estimate of $p(\mathbf{v}_U|\mathbf{v}_O)$.

A high score in this test indicates both the realism of samples and the generalization capability of a model, since models that do not allocate probability mass on good shapes (from the 'true' generating distribution of horses) and models that waste probability mass on bad shapes are both penalized. In particular for the motorbike dataset we found a small amount of regularisation to be beneficial for most models. This prevented overly confident predictions (and hence large penalties in the log-probability), e.g. in the situation where a particular pixel happened to be 0 for all training images, but 1 in one or some of the test images. To this end we replaced the predicted probability p of a pixel being 1 given the observed portion of the image by $d + (1 - 2d) \cdot p$. The results of these experiments can be seen in Table 4.2. For optimal damping SBM is the top-performing model on both the horses and motorbikes datasets, but the FA model performs well on the motorbikes.

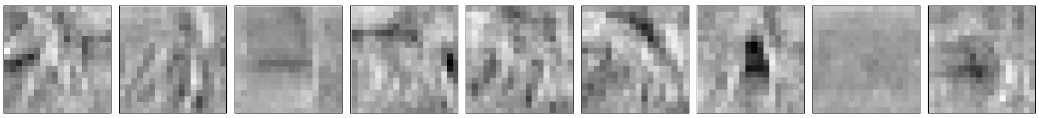
4.4.2 Analysis of the SBM formulation

So far we have demonstrated that the SBM is able to learn strong models of object shapes, producing realistic samples without overfitting to the training data. In this section we explore in more detail how these capabilities of the SBM depend on the specific properties of the architecture described in Sec. 4.2: local receptive field and weight sharing; hierarchical formulation; and receptive field overlap.

[2]We set $S = 10,000$ in our experiments.

(a) Factor Analysis

(b) Restricted Boltzmann Machine



(c) Shape Boltzmann Machine

Figure 4.14: **First layer example weights**. (a) Weights learned by the FA model capture only global modes of variability (32×32). (b) Weights learned by the RBM also fail to capture local modes of variation (32×32). (c) General, more local filters learned by an SBM (18×18).

		Horses		**Motorbikes**	
		Score	d	Score	d
Without regularisation	Mean	-50.72	0.000	-248.28	0.000
	FA	-41.28	0.000	**-109.17**	0.000
	RBM	-48.57	0.000	-142.47	0.000
	SBM	**-27.90**	0.000	-132.97	0.000
With regularisation	Mean	-50.65	0.012	-154.14	0.010
	FA	-40.33	0.028	-108.41	0.006
	RBM	-47.52	0.016	-142.47	0.000
	SBM	**-26.90**	0.014	**-104.21**	0.034

Table 4.2: **Imputation scores.** In the 'with regularisation' scenario, we also report for each model the regularisation d which maximizes that model's score.

4.4.2.1 Generalisation through local receptive fields

In the first layer of the SBM we employ localised receptive fields and parameter sharing. This dramatically reduces the number of parameters that need to be learned and in consequence substantially reduces the propensity of the model to overfit.

One way to diagnose this effect is to inspect the first layer weight matrix of the SBM and compare it to those of the two baseline models (RBM and FA) which were implemented without weight sharing. Each column in the weight matrices W of the models (Eqs. 4.5, 4.13, 4.21 for the RBM, SBM, and FA model respectively) corresponds to a 'filter' that is associated with the activation of one of the hidden units. As shown in Fig. 4.14a and 4.14b, the filters for the FA and RBM have only global structure. This means that these models are unable to combine local filters to generate novel horse shapes. In contrast, because spatial locality and parameter-sharing are built into the SBM, it learns general-purpose filters that allow it to generalise factorially from the training examples as can be seen in Fig. 4.14c.

Increasing the number of hidden units in the RBM in the hope that additional capacity would allow it to learn more local filters did not solve the problem but rather worsened the overall results, suggesting that it is indeed the lack of data rather than a lack of capacity that is the issue. On the other hand, an RBM with similar connectivity constraints as the first layer of the ShapeBM has fewer parameters than a fully con-

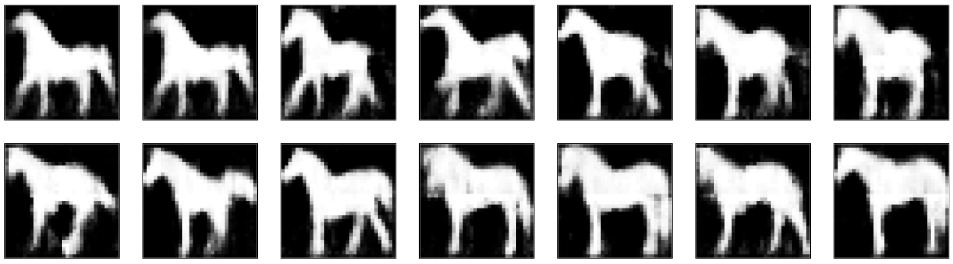
nected RBM and thus suffers less from overfitting (cf. Fig. 4.15). But as we discuss in more detail in the next section, without the second layer it fails to account for global constraints on the shape.

4.4.2.2 Global consistency through hierarchy

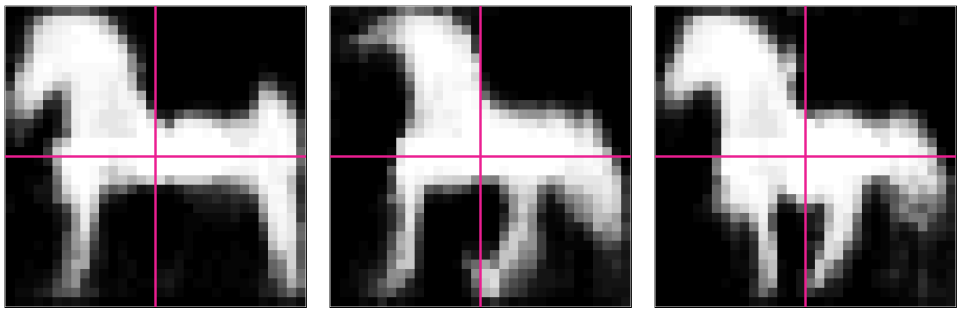
Localised receptive fields and weight sharing are crucial for the ability of the SBM to generalise well. In order to obtain a model that produces realistic samples these need to be embedded in a hierarchical architecture that ensures the global consistency of the shapes.

This is demonstrated by the samples in Fig. 4.15. They are obtained from an RBM equivalent to only the first layer of the SBM, i.e. this RBM has localised receptive fields with a small overlap between them. It was trained on the Weizmann horse dataset and has the same number of hidden units as the first layer of the horse SBM for which we have shown samples above. Unlike the fully connected RBM whose samples are shown in Fig. 4.7c this constrained RBM learns to generate a diverse set of shapes. The samples are, however, only locally plausible. In contrast to the samples from the SBM they do not exhibit any of the large-scale structure present in the training data and therefore are not realistic horse shapes in most cases. The second layer of the SBM is crucial for enforcing global consistency of the shapes.

In order to further understand the role of the hierarchy and to tease apart the roles of the two layers of the SBM in representing shape information we performed the following experiment: we fixed the configuration of the hidden units in the second layer ($\mathbf{h}^2$) to values inferred from two training images and then iterated between sampling $\mathbf{v}$ and $\mathbf{h}^1$ only. In Fig. 4.16 we plot two sets of samples for two different settings of $\mathbf{h}^2$. We observe that by freezing $\mathbf{h}^2$ we fix the horse's pose, but since $\mathbf{h}^1$ changes from sample to sample the position of its legs and other small details vary. This suggests that the highest layer in the model predominantly captures global information and has learned to be *invariant* to small-scale changes in shape (achieving an effect similar to the pooling layers e.g. in Lee et al., 2009). This automatic, implicit, separation of large-scale and small-scale statistics is fundamental to the operation of the model.



(a) Samples



(b) Global errors

Figure 4.15: **Samples from an SBM with only a single layer**. (a) A set of samples drawn from an RBM with the same connectivity constraints (localised receptive fields; small receptive field overlap; weight sharing) as the first layer of the SBM. Although the RBM enforces local smoothness (including at the receptive field boundaries, due to the overlap) it fails to enforce global constraints on the pose of the horses therefore often appears distorted (see, in particular, examples in (b); the pink lines indicate receptive field boundaries). Note that the visible biases b_i are *not* shared, and this is what allows the model to reproduce very coarsely the main features of horse shapes.

Figure 4.16: **Clamped sampling**. Sampling chains are run for two fixed, but different, configurations of $\mathbf{h}^2$. The horse's pose remains fixed, but configurations of legs, and neck and back positions vary. This suggests that the highest layer in the model predominantly captures high-level pose information.

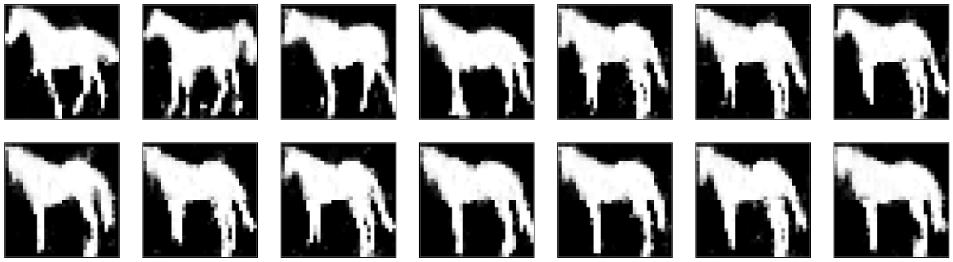
4.4.2.3 Local consistency through receptive field overlap

The hierarchical formulation encourages *global* consistency of the shapes by coordinating the overall pose across receptive fields. In order to also ensure *local* consistency at the receptive field boundaries we further introduced a small overlap of the receptive fields (denoted by r in Fig. 4.6).

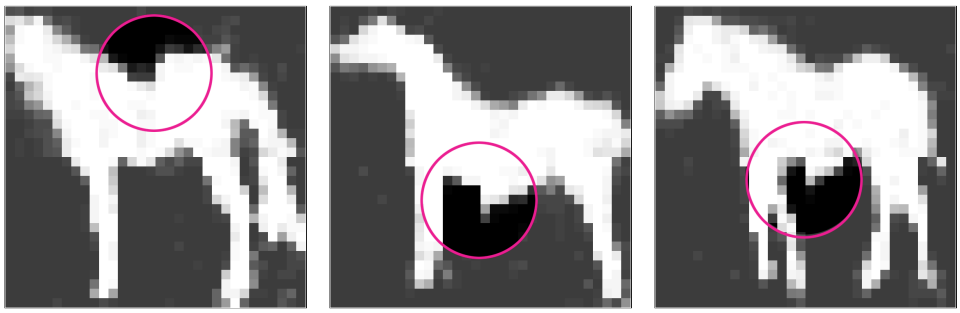
Empirically we found that this small overlap substantially improved model quality. The effect of this is illustrated in Fig. 4.17 where we show samples from an SBM (2-layer with local receptive fields and weight sharing) trained in the usual manner, except that there is *no* receptive field overlap (i.e. $r = 0$). This leads to a loss of continuity at the patch boundaries and also (albeit to a lesser extent) to a more global deterioration of sample quality, suggesting that the second layer on its own struggles to enforce local consistency. This global deterioration is due to the fact that some of the modelling capacity of the second layer is now needed to enforce local continuity. Increasing the number of hidden units in the *second* layer would reduce this deterioration at the cost of increasing the number of parameters and so reducing the advantage gained from the hierarchical structure. Experimentally we found that it led to overfitting and did not give satisfactory results.

4.4.3 Multiple object categories

Class-specific shape models are appropriate if the class is known, but for segmentation/detection applications this may not be the case. A similar situation arises if the



(a) Samples



(b) Misalignments

Figure 4.17: **Samples without overlap**. (a) Samples from a SBM trained on Weizmann horses in the same way as the SBM described in Sec. 4.4.1 except that there is no receptive field overlap in the first layer (i.e. $r = 0$). The lack of receptive field overlap leads to discontinuities at the receptive field boundaries not present in the samples from the SBM trained with $r = 4$ (see in particular the examples highlighted in (b) and compare to the SBM samples shown in Fig. 4.7d) and more generally reduces the overall sample quality somewhat.

view point is not fixed (e.g. objects can appear right or left facing). In both cases there is large overall variability in the data but the data also form relatively distinct clusters of similar shapes (e.g. all objects from a particular category, or all right-facing objects).

To investigate whether the SBM is able to successfully deal with such additional variability and structure in the data we applied it to a dataset consisting of shapes from multiple object classes and tested whether it would be able to learn a strong model of the shapes of all classes simultaneously.

We trained an SBM on a combination of the Weizmann data and 3 other animal categories from Caltech-101 (Fei-Fei et al., 2004). In addition to 327 horse images, the dataset contains images of 68 dragonflies, 78 llamas and 59 rhinos (for a total of 531 images). The images are cropped and normalised to 32×32 pixels. An SBM with $r = 4$, and 2,000 and 400 units for $\mathbf{h}^1$ and $\mathbf{h}^2$ respectively was jointly trained *without* information about image class.

In our experiments we found that the SBM still learns a strong model, as demonstrated by Fig. 4.18 which shows samples as well as shape completions obtained from the learned model.

We further wanted to know whether the SBM's unsupervised learning procedure has led it to discover the underlying grouping of the shapes into categories. In order to test this, we compute average inter- and intra-class distances of all training instances, both in data-space and in latent-space[3]. We then inspect the ratios of these distances for the two representations.

More precisely, for each object class c, let $\mathcal{D}_c$ be the set images belonging to class c, and $\mathcal{D}_{\setminus c}$ be the set of images from all other classes, then

$$d_c^{\text{inter}} = \sum_{\mathbf{v}^m \in \mathcal{D}_c} \sum_{\mathbf{v}^n \in \mathcal{D}_{\setminus c}} \sqrt{(\mathbf{v}^m - \mathbf{v}^n)^2} / N^{\text{inter}}, \tag{4.23}$$

$$d_c^{\text{intra}} = \sum_{\mathbf{v}^m \in \mathcal{D}_c} \sum_{\mathbf{v}^n \in \mathcal{D}_c} \sqrt{(\mathbf{v}^m - \mathbf{v}^n)^2} / N^{\text{intra}}, \tag{4.24}$$

and

$$\text{ratio}_c = \frac{d_c^{\text{inter}}}{d_c^{\text{intra}}}. \tag{4.25}$$

Values of ratio_c that are greater than 1 indicate that inter-class distances are larger. In Fig. 4.19a we plot these ratios for the four classes. These results suggest that the SBM

[3]We use $\mathbf{v}$ as an instance's representation in data-space, and the values $\mathbf{h}^2$ of the mean-field approximation to the posterior $p(\mathbf{h}^1, \mathbf{h}^2|\mathbf{v})$ as its representation in latent-space.

latent representation groups the shapes from each category much more closely than they are in pixel-space.

We also tested how well the model discovered object categories by using it to classify in a setting with few labelled examples. We trained a generalised linear model (GLM) using the `glmnet` algorithm (Friedman et al., 2010) on between $T = 1 \dots 20$ randomly selected images of each category and tested on $59 - T$ images per category, averaging over 100 runs. We find that despite its smaller size, given only a few training examples, the latent $\mathbf{h}^2$ is most discriminative (see Fig. 4.19b). After just one labelled example per category, classification accuracy using the trained GLMs is 56.0% using $\mathbf{h}^2$ vs. just 36.8% using $\mathbf{v}$.

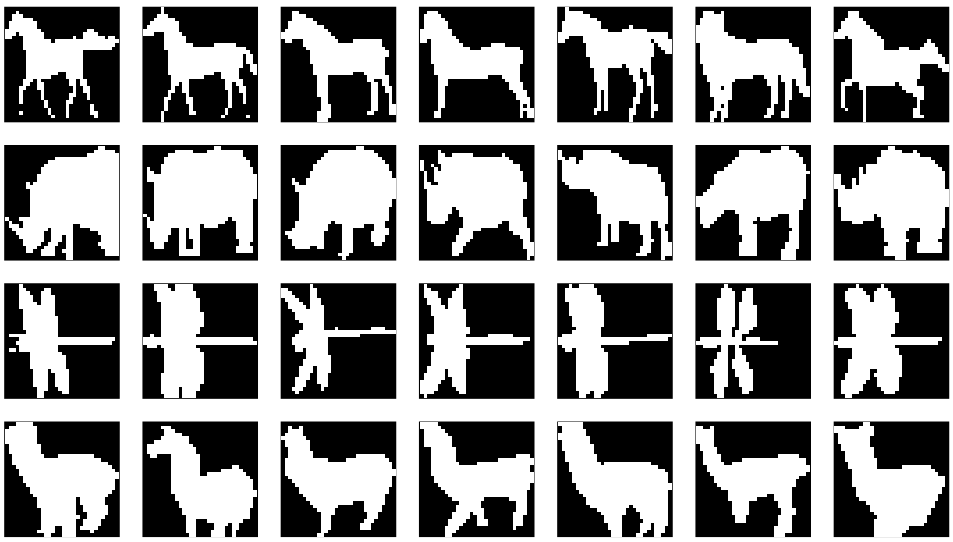
Overall these results suggest that the SBM is not only able to deal with the additional variability arising from multiple object classes, but also reliably generalises within each class. It further appears to naturally separate clusters of related shapes in its latent representation, which can be exploited, for instance, for classification purposes.

4.5 Conclusions

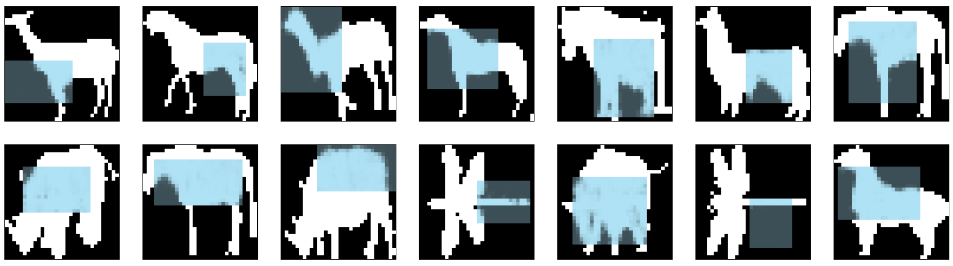
In this chapter we have presented the Shape Boltzmann Machine, a strong generative model of object shape. The SBM is based on the general DBM architecture, a form of undirected graphical model that makes heavy use of latent variables to model high-order dependencies between the observed variables. We believe that the *combination* of (a) carefully chosen connectivity and capacity constraints, along with (b) a hierarchical architecture, and (c) a training procedure that allows for the joint optimisation of the full model, is key to the success of the SBM.

These ingredients allow the SBM to learn high quality probability distributions over object shapes from small datasets, consisting of just a few hundred training images. The learned models are convincing in terms of both realism of samples from the distribution and generalisation to new examples of the same shape class. Without making use of specialist knowledge about the particular shapes the model develops a natural representation with some division of labour across layers.

The SBM can also directly be used as a component of a more comprehensive probabilistic architecture. As demonstrated in Le Roux et al. (2011), and Heess et al. (2011), for instance, it is possible to combine undirected models of shapes formulated as RBMs



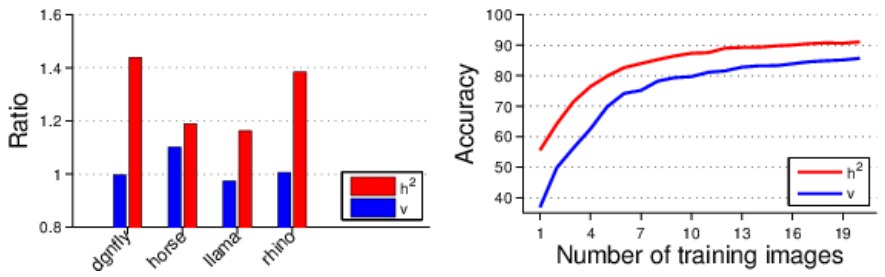
(a) Training



(b) Sampled completions

(c) Samples

Figure 4.18: **Multiple object categories**. (a) A selection of images from the augmented dataset. (b) The model simultaneously identifies the object class and fills in the missing image region (shaded blue). (c) Samples from a single tempered chain.



(a) (b)

Figure 4.19: **Classification using the learned representation**. (a) The ratio of inter- and intra-class distances (values > 1 indicate that inter-class distances are larger). (b) GLM classification accuracy as a function of the number of training images, averaged over 100 runs.

or DBMs with models of appearance to obtain complete probabilistic generative models of RGB images with well-defined and efficient inference schemes. Such models allow reasoning about various image properties and can be applied, for instance, to segmentation tasks (Heess et al., 2011).

In Chapter 5 we show how a multi-region SBM can be integrated into the generative model of images we described in Chapter 3, FSA, and demonstrate that it can be used to obtain competitive results on two challenging parts-based segmentation benchmarks.

Chapter 5

A Boltzmann Machine Model for Parts-based Object Segmentation

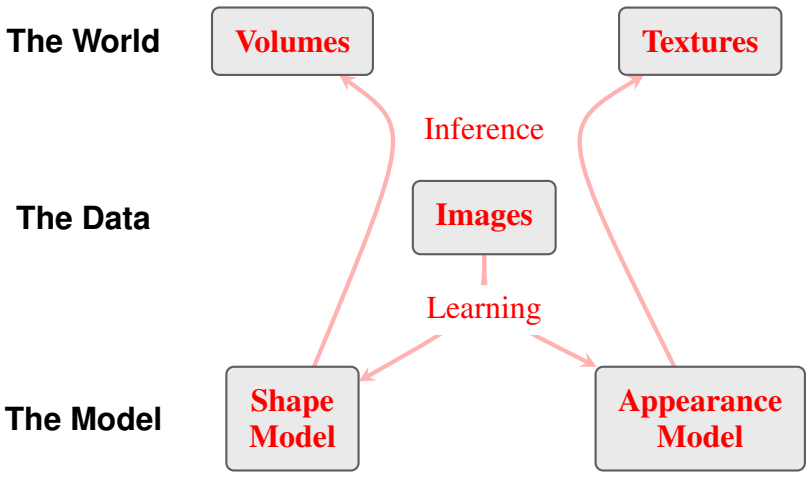


Figure 5.1: Inference in fully generative models of images of objects.

In Chapter 3 we presented FSA, a generative probabilistic model of objects that learns about the shapes and appearances of their parts. The performance of FSA on the object segmentation task generally depends on its ability to learn accurate models of shapes and appearances from training data, especially when it is applied to datasets that exhibit large amounts of variability. In general, the stronger the models of these two components, the more performance should be improved.

In Chapter 4 we presented a generative probabilistic model of binary object shapes, the Shape Boltzmann Machine (SBM). We demonstrated that the SBM constitutes the state-of-the-art and it possesses several highly desirable characteristics: samples from

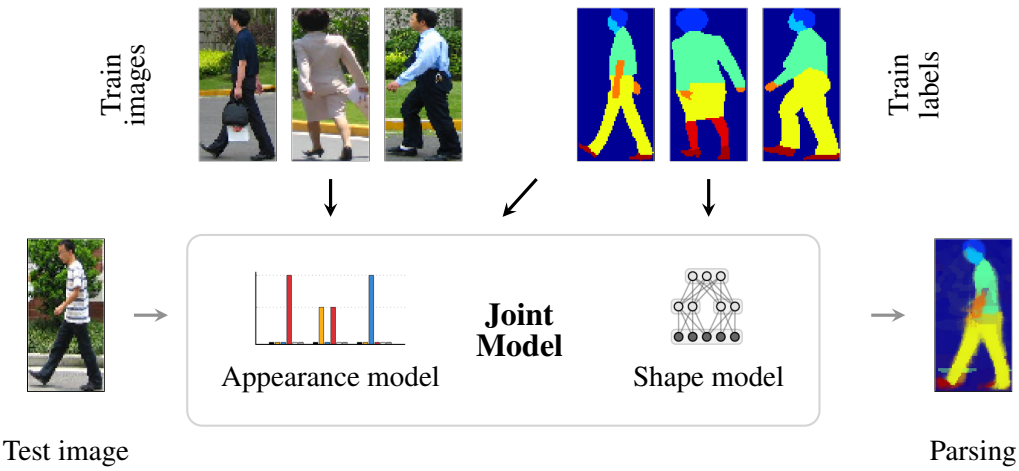


Figure 5.2: **Training and testing overview**. Using annotated images separate models of shape and appearance are trained. Given an unseen test image, its parsing is obtained via inference in the proposed joint model.

the model look realistic, and it generalises to generate samples that differ from the limited number of examples it is trained on.

In this chapter we consider whether FSA's structure can be used in conjunction with the SBM to form a strong parts-based model of images of objects:

First, in order to account for object parts we extend the SBM to use multinomial visible units instead of binary ones, resulting in the Multinomial Shape Boltzmann Machine (MSBM), and we demonstrate that the MSBM too constitutes a strong model of parts-based object shape.

We then combine the MSBM with an appearance model, similar to the way it is done in FSA, to form a fully generative model of images of objects. We show how parts-based object segmentations can be obtained simply by performing probabilistic inference in this joint model (see Fig. 5.2). We finally apply our model to two challenging datasets and find that in addition to being fully generative, the model's performance is comparable to the state-of-the-art.

The remainder of this chapter is structured as follows: In Secs. 5.1 and 5.2 we present the model and propose efficient inference and learning schemes. In Sec. 5.3 we compare and contrast the resulting joint model with existing work in the literature. We describe our experimental results in Sec. 5.4 and conclude with a discussion in Sec. 5.5.

5.1 Model

As in Chapter 3, we consider datasets of cropped images of an object class. We assume that the images are constructed through some combination of a fixed number of parts. Given a dataset $\mathbf{D} = \{\mathbf{X}^d\}, d = 1...n$ of such images $\mathbf{X}$, each consisting of P pixels $\{\mathbf{x}_i\}, i = 1...P$, we wish to infer a segmentation $\mathbf{S}$ for the image. $\mathbf{S}$ consists of a labelling $\mathbf{s}_i$ for every pixel, where $\mathbf{s}_i$ is a 1-of-$(L+1)$ encoded variable, and L is the fixed number of parts that combine to generate the foreground. In other words, $\mathbf{s}_i = (s_{li})$, $l = 0...L$, $s_{li} \in \{0, 1\}$ and $\sum_l s_{li} = 1$. As before the background is also treated as a 'part' ($l = 0$).

5.1.1 Part shapes

As discussed in Chapter 4, several types of models can be used to define probabilistic distributions over segmentations $\mathbf{S}$. The simplest approach is to model each pixel $\mathbf{s}_i$ independently with categorical variables whose parameters are specified by the object's mean shape (Fig. 5.3a). Markov random fields (MRFs, Fig. 5.3b) additionally model interactions between nearby pixels using potential functions that typically only capture local properties of images like smoothness and continuity. In Chapter 3, we used a latent Gaussian model of shape to account for global deformations of the object's shape (see Sec. 3.1.1 for further details).

In Chapter 4, we demonstrated that the Shape Boltzmann Machine (SBM) can be used to accurately capture the properties of shapes, that samples from the model look realistic, and that it generalises to generate samples that differ from the limited number of examples it is trained on.

The SBM represents shapes as binary images and can be used, for example, as a prior when segmenting a foreground object from its background. While it is often sufficient to consider the foreground object as a single region without internal structure, there are situations where it is desirable to explicitly model multiple, dependent regions, e.g. in order to decompose the foreground object into parts (e.g. the FSA model described in Chapter 3, also Winn and Jojic, 2005; Kapoor and Winn, 2006; Thomas et al., 2009; Bo and Fowlkes, 2011) which in turn, for example, can be used by a robot to interact with the object.

Here, we extend the SBM to account for multi-part shapes to obtain the Multinomial Shape Boltzmann Machine (MSBM). In the MSBM this can be achieved by using categorical visible units instead of binary ones. Visible units with $L+1$ different states (i.e. $\mathbf{s}_i \in \{0, \dots L\}$) allow the modelling of shapes with L parts.

As in the SBM, the MSBM has two layers of latent variables: $\mathbf{h}^1$ and $\mathbf{h}^2$ (collectively $\mathbf{H} = \{\mathbf{h}^1, \mathbf{h}^2\}$), and defines a Boltzmann distribution over segmentations:

$$p(\mathbf{S}) = \frac{1}{Z(\boldsymbol{\theta}^s)} \sum_{\mathbf{h}^1, \mathbf{h}^2} \exp\{-E(\mathbf{S}, \mathbf{h}^1, \mathbf{h}^2 | \boldsymbol{\theta}^s)\}, \tag{5.1}$$

except now

$$\begin{aligned} E(\mathbf{S}, \mathbf{h}^1, \mathbf{h}^2 | \boldsymbol{\theta}^s) = &\sum_{i,l} b_{li} s_{li} + \sum_{i,j,l} w^1_{lij} s_{li} h^1_j + \sum_j c^1_j h^1_j \\ &+ \sum_{j,k} w^2_{jk} h^1_j h^2_k + \sum_k c^2_k h^2_k, \end{aligned} \tag{5.2}$$

where l ranges over parts, j and k range over the first and second layer hidden variables, and $\boldsymbol{\theta}^s = \{W^1, W^2, \mathbf{b}, \mathbf{c}^1, \mathbf{c}^2\}$ are the shape model parameters.

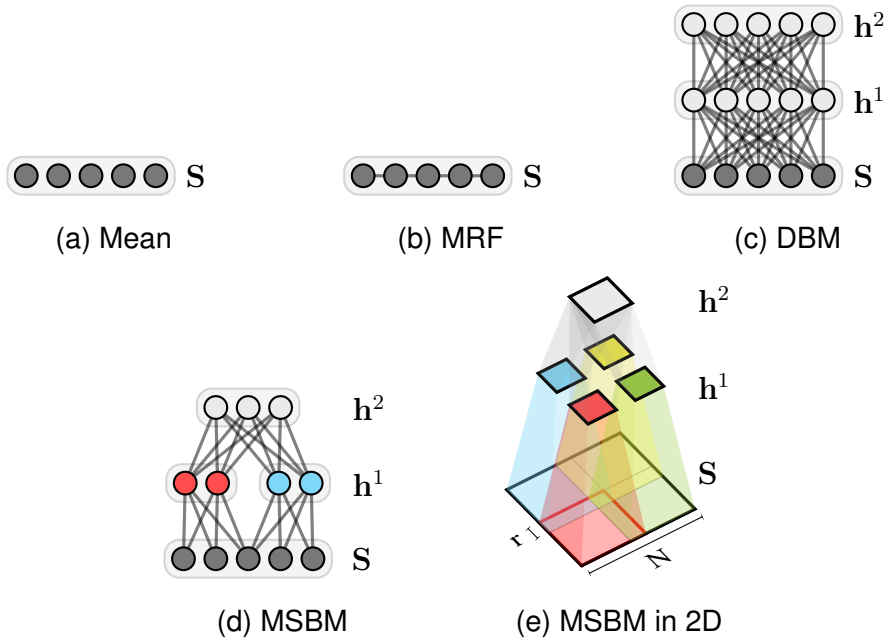
The structure remains the same as that of the SBM: In the first layer, local receptive fields are enforced by connecting each hidden unit in $\mathbf{h}^1$ only to a subset of the visible units, corresponding to one of four patches, as shown in Figs. 5.3d and 5.3e. Each patch overlaps its neighbour by r pixels, which allows boundary continuity to be learned at the lowest layer. We share weights between the four sets of first-layer hidden units and patches, and purposely restrict the number of units in $\mathbf{h}^2$. These modifications significantly reduce the number of parameters whilst taking into account an important property of shapes, namely that the strongest dependencies between pixels are typically local.

The change in the nature of the visible units preserves all of the appealing properties of the SBM. In particular the conditional distributions over the three sets of variables $\mathbf{S}$, $\mathbf{h}^1$, and $\mathbf{h}^2$ remain factorial. The only change is in the specific forms of the two conditional distributions $p(\mathbf{S}|\mathbf{h}^1)$ and $p(\mathbf{h}^1|\mathbf{S}, \mathbf{h}^2)$:

$$p(\mathbf{s}_i = l | \mathbf{h}^1) = \frac{\exp(\sum_j w^1_{lij} h^1_j + b_{li})}{\sum_{l'=0}^{L} \exp(\sum_j w^1_{l'ij} h^1_j + b_{l'i})}, \tag{5.3}$$

$$p(h^1_j = 1 | \mathbf{S}, \mathbf{h}^2) = \sigma(\sum_{i,l} w^1_{lij} s_{li} + \sum_k w^2_{jk} h^2_k + c^1_j), \tag{5.4}$$

where in the left-hand-side of Eq. (5.3) we use $\mathbf{s}_i = l$ to denote the fact that $s_{li} = 1$ and $s_{l'i} = 0, \ \forall l' \neq l$ as explained above.



(a) Mean (b) MRF (c) DBM

(d) MSBM (e) MSBM in 2D

Figure 5.3: **Models of shape**. Object shape is modelled with undirected graphical models. (a) 1D slice of a mean model. (b) Markov random field in 1D. (c) Deep Boltzmann Machine in 1D. (d) 1D slice of a Shape Boltzmann Machine. (e) Shape Boltzmann Machine in 2D. In all models latent units $\mathbf{h}$ are binary and visible units $\mathbf{S}$ are multinomial random variables.

Note that Eq. 5.4 is effectively the same as Eq. 4.17 except that there are now $L+1$ binary visible units per pixel. Consequently, at each visible location i and hidden location j, the model now has parameters w_{lij} and b_{li} – one of each for the $L+1$ different states, whereas in the binary case it only had the w_{ij} and b_i.The conditional distribution given in Eq. 5.3 implements the constraint that for each pixel only one of these $L+1$ binary units can be active, i.e. only one of the parts can be present. Due to the particular form of the conditional distribution (Eq. 5.3) categorical visible units are often referred to as 'softmax' units (e.g. Bridle, 1990). In our experiments below we explore SBMs with 6 or 7 parts.

It should be noted that the above formulation of the multi-part MSBM is especially suited to model the shapes of several *dependent* regions such as non-occluding (or lightly occluding) object parts. For modelling the shapes of multiple *independent* regions, as arise in the case of multiple occluding objects, it might be more suitable to model occlusion explicitly, as in Le Roux et al. (2011).

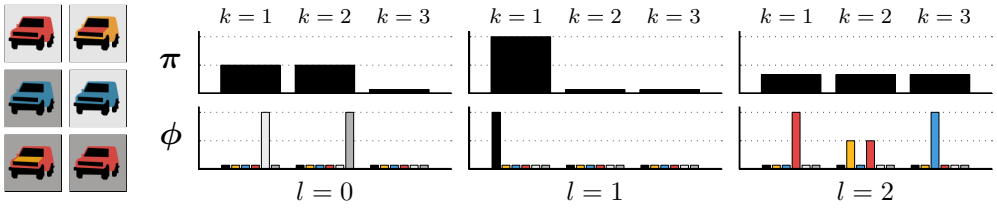


Figure 5.4: **Appearance modelling using mixtures of histograms**. *Left:* An exemplar dataset. Here we assume one background ($l = 0$) and two foreground ($l = 1$, non-body; $l = 2$, body) parts. *Right:* The corresponding appearance model. In this example, $L = 2$, $K = 3$ and $W = 6$. Best viewed in colour.

5.1.2 Part appearances

As with FSA, pixels in a given image are assumed to have been generated by W fixed Gaussians in RGB space. During pre-training, the means $\{\boldsymbol{\mu}_w\}$ and covariances $\{\boldsymbol{\Sigma}_w\}$ of these Gaussians are extracted by training a mixture model with W components on every pixel in the dataset, ignoring image and part structure. It is also assumed that each of the L parts can have different appearances in different images, and that these appearances can be clustered into K classes per part. The classes differ in how likely they are to use each of the W components when 'colouring in' the part.

The generative process remains unchanged from FSA. For part l in an image, one of the K classes is chosen (represented by a 1-of-K indicator variable $\mathbf{a}_l$). Given $\mathbf{a}_l$, the probability distribution defined on pixels associated with part l is given by a Gaussian mixture model with means $\{\boldsymbol{\mu}_w\}$ and covariances $\{\boldsymbol{\Sigma}_w\}$ and mixing proportions $\{\phi_{lkw}\}$. The prior on $\mathbf{A} = \{\mathbf{a}_l\}$ specifies the probability π_{lk} of appearance class k being chosen for part l. Therefore appearance parameters $\boldsymbol{\theta}^a = \{\pi_{lk}, \phi_{lkw}\}$. See Fig. 5.4 for an illustration, and Sec. 3.1.2 for further details.

5.1.3 Combining shapes and appearances

The latent variables for image $\mathbf{X}$ are $\mathbf{A}$, $\mathbf{S}$, $\mathbf{H}$, and the model's active parameters $\boldsymbol{\theta}$ include shape parameters $\boldsymbol{\theta}^s$ and appearance parameters $\boldsymbol{\theta}^a$, so that

$$p(\mathbf{X}, \mathbf{A}, \mathbf{S}, \mathbf{H}|\boldsymbol{\theta}) = \frac{1}{Z(\lambda)} p(\mathbf{A}|\boldsymbol{\theta}^a) p(\mathbf{S}, \mathbf{H}|\boldsymbol{\theta}^s) \prod_{i=1}^{P} p(\mathbf{x}_i|\mathbf{A}, \mathbf{s}_i, \boldsymbol{\theta}^a)^{\lambda}. \quad (5.5)$$

Algorithm 1 MCMC inference algorithm

procedure INFER($\mathbf{X}, \boldsymbol{\theta}$)
  Initialise $\mathbf{S}^1$, $\mathbf{H}^1$
  for $t \leftarrow 2 : \mathit{chain\_length}$ **do**
    $\mathbf{A}^t \sim p(\mathbf{A}|\mathbf{S}^{t-1}, \mathbf{H}^{t-1}, \mathbf{X}, \boldsymbol{\theta})$
    $\mathbf{S}^t \sim p(\mathbf{S}|\mathbf{A}^t, \mathbf{H}^{t-1}, \mathbf{X}, \boldsymbol{\theta})$
    $\mathbf{H}^t \sim p(\mathbf{H}|\mathbf{S}^t, \boldsymbol{\theta})$
  end for
end procedure
return $\{\mathbf{S}^t\}_{t=\mathit{burnin}:\mathit{chain\_length}}$

Note that here we introduce the parameter λ to adjust the relative contributions of the shape and appearance components (cf. its use in speech recognition, e.g. Bahl et al., 1980). The effect is similar to that produced by the 'leak' parameter in FSA (Eq. 3.2). Also note that $Z(\lambda)$ is constant throughout the execution of the algorithms. We set λ via trial and error in our experiments.

See Fig. 5.5 for an illustration of the complete graphical model. During learning, we find the values of $\boldsymbol{\theta}$ that maximise the likelihood of the training data $\mathbf{D}$, and segmentation is performed on a previously-unseen image by querying the marginal distribution $p(\mathbf{S}|\mathbf{X}^{\text{test}}, \boldsymbol{\theta})$.

5.2 Inference and learning

5.2.1 Inference

As with FSA, we approximate $p(\mathbf{A}, \mathbf{S}, \mathbf{H}|\mathbf{X}, \boldsymbol{\theta})$ by drawing samples of $\mathbf{A}$, $\mathbf{S}$ and $\mathbf{H}$ using block-Gibbs Markov Chain Monte Carlo (MCMC). The desired distribution $p(\mathbf{S}|\mathbf{X}, \boldsymbol{\theta})$ can be obtained by considering only the samples for $\mathbf{S}$ (see Algorithm 1). In order to sample $p(\mathbf{A}|\mathbf{S}, \mathbf{H}, \mathbf{X}, \boldsymbol{\theta})$, we consider the conditional distribution of appearance class k being chosen for part l which is given by

$$p(a_{lk} = 1|\mathbf{S}, \mathbf{X}, \boldsymbol{\theta}) = \frac{\pi_{lk} \prod_i \left(\sum_w \phi_{lkw} \mathcal{N}(\mathbf{x}_i|\boldsymbol{\mu}_w, \boldsymbol{\Sigma}_w)\right)^{\lambda \cdot s_{li}}}{\sum_{r=1}^{K} \left[\pi_{lr} \prod_i \left(\sum_w \phi_{lrw} \mathcal{N}(\mathbf{x}_i|\boldsymbol{\mu}_w, \boldsymbol{\Sigma}_w)\right)^{\lambda \cdot s_{li}}\right]}. \tag{5.6}$$

Since the MSBM only has edges between each pair of adjacent layers, all hidden units

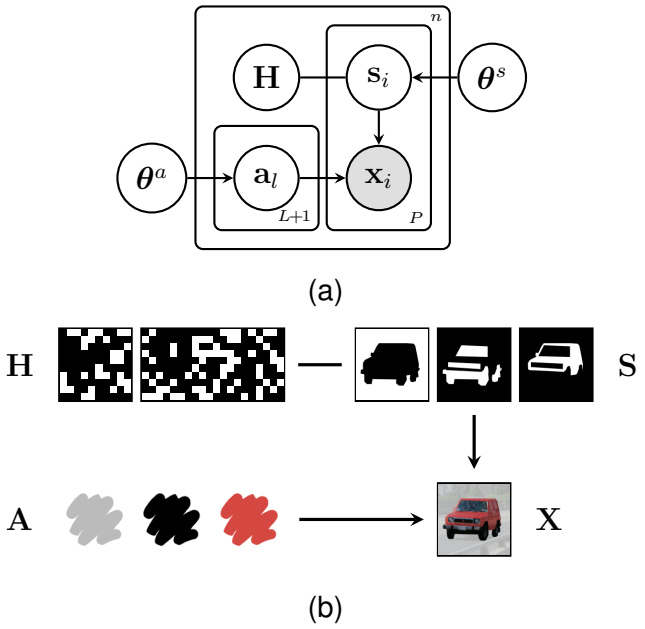


Figure 5.5: **A model of shape *and* appearance**. (a) The joint model. Pixels $\mathbf{x}_i$ are modelled via appearance variables $\mathbf{A}_l$. The model's belief about each layer's shape is captured by shape variables $\mathbf{H}$. Segmentation variables $\mathbf{s}_i$ assign each pixel to a layer. (b) Schematic for an image $\mathbf{X}$.

within a layer are conditionally independent given the units in the other two layers. This property can be exploited to make inference in the shape model exact and efficient. The conditional probabilities are:

$$p(h_j^1 = 1|\mathbf{S}, \mathbf{h}^2, \boldsymbol{\theta}) = \sigma(\sum_{i,l} w_{lij}^1 s_{li} + \sum_k w_{jk}^2 h_k^2 + c_j^1), \tag{5.7}$$

$$p(h_k^2 = 1|\mathbf{h}^1, \boldsymbol{\theta}) = \sigma(\sum_j w_{jk}^2 h_j^1 + c_j^2), \tag{5.8}$$

where $\sigma(y) = 1/(1+\exp(-y))$ is the sigmoid function. To sample from $p(\mathbf{H}|\mathbf{S}, \mathbf{X}, \boldsymbol{\theta})$ we iterate between Eqs. 5.7 and 5.8 multiple times and keep only the final values of $\mathbf{h}^1$ and $\mathbf{h}^2$. Finally, we draw samples for the pixels in $p(\mathbf{S}|\mathbf{A}, \mathbf{H}, \mathbf{X}, \boldsymbol{\theta})$ independently:

$$p(s_{li} = 1|\mathbf{A}, \mathbf{H}, \mathbf{X}, \boldsymbol{\theta}) = \frac{\exp(\sum_j w_{lij}^1 h_j^1 + b_{li})\, p(\mathbf{x}_i|\mathbf{A}, s_{li} = 1, \boldsymbol{\theta})^\lambda}{\sum_{m=1}^L \exp(\sum_j w_{mij}^1 h_j^1 + b_{mi})\, p(\mathbf{x}_i|\mathbf{A}, s_{mi} = 1, \boldsymbol{\theta})^\lambda}. \tag{5.9}$$

5.2.2 Seeding

The latent-space in this model is extremely high-dimensional and inference chains often get stuck. This can mainly be attributed to the block-Gibbs sampling procedure which fails to find probable settings of $\mathbf{A}$ and $\mathbf{S}$ for a given image $\mathbf{X}$, since any local move in the space of appearances or the space of shapes would lead to a less probable inference for that image.

In practice we find it helpful to run several inference chains, each initialising $\mathbf{S}^1$ to a different value. It is then necessary to devise a scheme through which the 'best' inference is retained and the others are discarded. The computation of the likelihood $p(\mathbf{X}|\boldsymbol{\theta})$ of image $\mathbf{X}$ is intractable, so we approximate the quality of each inference using a scoring function:

$$\text{Score}(\mathbf{X}|\boldsymbol{\theta}) = \frac{1}{T}\sum_t p(\mathbf{X}, \mathbf{A}^t, \mathbf{S}^t, \mathbf{H}^t, \boldsymbol{\theta}), \tag{5.10}$$

where $\{\mathbf{A}^t, \mathbf{S}^t, \mathbf{H}^t\}, t = 1...T$ are the samples obtained from the posterior distribution $p(\mathbf{A}, \mathbf{S}, \mathbf{H}|\mathbf{X}, \boldsymbol{\theta})$. Although this estimator is biased, we find that it works well in practice. See e.g. Heess, 2011, p. 107-109 for further discussion of this issue.

5.2.3 Learning

Learning of the model involves maximising the log likelihood $\log p(\mathbf{D}|\boldsymbol{\theta}^a, \boldsymbol{\theta}^s)$ of the training dataset $\mathbf{D}$ with respect to λ, the appearance parameters $\boldsymbol{\theta}^a$ and the shape parameters $\boldsymbol{\theta}^s$. Since training is partially supervised, in that for each image $\mathbf{X}$ its corresponding segmentation $\mathbf{S}$ is also given, we can learn the parameters of the shape and appearance components separately. For appearances, the learning of the mixing coefficients and the histogram parameters decomposes into mixture updates independently for each part, exactly as in FSA (see Chapter 3), and for shapes, the learning is structurally identical to that of the SBM (see Chapter 4).

5.3 Related work

Existing probabilistic models of images can be categorised by the amount of variability they expect to encounter in the data and by how they model this variability. A significant portion of the literature models images using only two parts: a foreground object

and its background (e.g. Rother et al., 2004; Borenstein et al., 2004; Arora et al., 2007; Alexe et al., 2010). Models that account for the parts within the foreground object mainly differ in how accurately they learn about and represent the variability of the shapes of the object's parts.

In Probabilistic Index Maps (PIMs, Jojic and Caspi, 2004) a mean partitioning is learned, and the deformable PIM (Winn and Jojic, 2005) additionally allows for local deformations of this mean partitioning. Stel Component Analysis (Jojic et al., 2009) accounts for larger amounts of shape variability by learning a number of different template means for the object that are blended together on a pixel-per-pixel basis, and FSA (Chapter 3) models global properties of shape using a latent-Gaussian model. However, we showed in Chapter 4 that none of these models constitute a strong model of shape in terms of realism of samples and generalisation capabilities. We will demonstrate in Sec. 5.4 that, like the SBM, the MSBM does in fact possess these properties.

The closest works to ours in terms of ability to deal with datasets that exhibit significant variability in both shape and appearance are the works of Bo and Fowlkes (2011) and Thomas et al. (2009). Bo and Fowlkes (2011) present an algorithm for pedestrian segmentation that models the shapes of the parts using several template means. The different parts are composed using hand coded geometric constraints, which means that the model cannot be automatically extended to other application domains. The Implicit Shape Model (ISM) used in Thomas et al. (2009) is reliant on interest point detectors and defines distributions over segmentations only in the *posterior*, and therefore is not fully generative. By contrast, the model presented here is entirely learned from data and fully generative, therefore it can be applied to new datasets and diagnosed with relative ease.

5.4 Experiments

In this section we analyse the learned shape and appearance components for two datasets of pedestrians and cars and evaluate the performance of the model on the foreground and parts-based object segmentation tasks.

5.4.1 Penn-Fudan pedestrians

The first dataset that we considered is Penn-Fudan pedestrians (Russell et al., 2008), consisting of 169 images of pedestrians (Fig. 5.7a). The images are annotated with ground-truth segmentations for $L = 7$ different parts (hair, face, upper and lower clothes, shoes, legs, arms). We compare the performance of the model with the algorithm of Bo and Fowlkes (Bo and Fowlkes, 2011).

For the shape component, we trained an SBM on the 684 images of a labelled version of the HumanEva dataset (Sigal et al., 2010) (at 48×24 pixels; also flipped horizontally) with overlap $r = 4$, and 400 and 50 hidden units in the first and second layers respectively. This dataset was also used by Bo and Fowlkes (2011) and was annotated by them. Each layer was pre-trained for 3000 epochs. After pre-training, joint training was performed for 1000 epochs. We also trained an FA shape model with $H = 5$ latent space dimensions for comparison (see Sec. 5.4.3 for details).

To assess the realism and generalisation characteristics of the learned MSBM we sample from it. In Fig. 5.7c we show a chain of unconstrained samples from an SBM generated via block-Gibbs MCMC (1000 samples between frames). Note that like the SBM, the MSBM captures highly non-linear correlations in the data whilst preserving the object's details (e.g. the pedestrians' faces and arms, unlike FA).

As before, to demonstrate that the model has not simply memorised the training data, we examine the differences between the sampled shapes in Fig. 5.7c and their closest images in the training set (based on per-pixel label agreement). We plot the results of this experiment in Fig. 5.7d. We see that the model generalises in non-trivial ways to generate *realistic* shapes that it had not encountered during training.

In Fig. 5.7e we show how the MSBM completes rectangular occlusions. The samples highlight the variability in possible completions captured by the model. Note how, e.g. the length of the person's trousers on one leg affects the model's predictions for the other, demonstrating the model's knowledge about long-range dependencies.

Overall these results demonstrate that the multi-part formulation of the MSBM significantly extends the binary SBM. The MSBM learns distributions over shapes with internal structure whilst preserving its ability to generalise and to produce realistic samples.

We then split the Penn-Fudan dataset (at 200×100 pixels) into 10 train/test cross-

validation splits without replacement. We used the training images in each split to train the appearance component with a vocabulary of size $W = 50$ and $K = 100$ mixture components.[1] We additionally constrained the model by sharing the appearance models for the arms and legs with that of the face.

We assessed the quality of the appearance model by performing the following experiment: For each test image, we used the scoring function described in Eq. 5.10 to evaluate a number of different proposal segmentations for that image. We considered 10 segmentations chosen randomly from the training dataset *as well* as the ground-truth segmentation for the test image, and found that the appearance model correctly assigns the highest score to the ground-truth 95% of the time. This result is reassuring as it suggests that the appearance model is sufficiently accurate to distinguish the ground truth segmentation from the others.

It is also possible to sample from the appearance model, and we do this in Fig. 5.6. The samples illustrate the kind of information captured by the appearance model. The 'upper' part (which corresponds to upper-body clothes) can appear in a wide variety of styles, but each style is likely to span only a limited part of colour space. Contrast this with the 'background' part which almost always has the same style of appearance, however this single style covers almost all colours. Also not how the 'face', 'arms' and 'legs' parts reflect the range of skin tones seen in the dataset.

During inference, the shape model and the response from the appearance model (which are defined on images of different sizes, the former at 48×24 and the latter at 200×100), were combined at 200×100 pixels via MATLAB's `imresize` function. We set $\lambda = 0.8$ (Eq. 5.9) via trial and error. We seeded inference chains at 100 exemplar segmentations from the HumanEva dataset (obtained using the K-medoids algorithm with $K = 100$), and ran them for 20 Gibbs iterations each (with 5 iterations of Eqs. 5.7 and 5.8 per Gibbs iteration). Our unoptimised MATLAB implementation completed inference for each chain in around 7 seconds.

In Table 5.1 we compute the conditional probability of each pixel belonging to different parts given the last set of samples obtained from the highest scoring chain, assign each pixel independently to the most likely part label at that pixel, and report the percentage of correctly labelled pixels.

[1] We obtained the best quantitative results with these settings. The appearances exhibited by the parts in the dataset are highly varied, and the complexity of the appearance model reflects this fact.

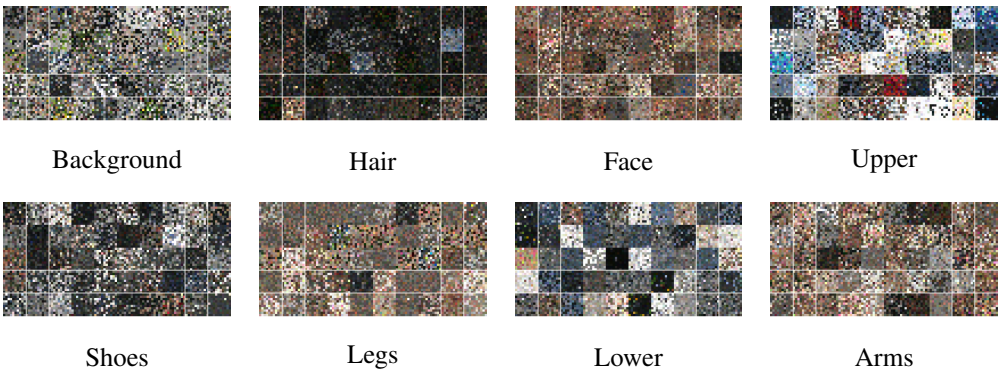


Figure 5.6: **Samples from the learned appearance model**. For each part l, one of the K classes is chosen based on the prior probabilities π_l. Given this choice, the probability distribution defined on pixels is given by mixing proportions ϕ_{lk}. 50 samples of size 10×10 pixels are then generated from these histograms for each part.

In our experiments we found that accuracy can be improved using superpixels (SP) computed on $\mathbf{X}$, so that pixels within a superpixel are all assigned the most common label within it. As with Bo and Fowlkes (2011) we use gPb-OWT-UCM (Arbelaez et al., 2009) to generate superpixels.

We also report the accuracy obtained, had the top scoring seed segmentation been returned as the final segmentation for each image. This allows us to determine whether MCMC inference provides any improvement over the segmentation provided by the seed. Here the quality of the seed is determined solely by the appearance model, and not by the shape model. We note that MCMC inference in the model significantly improves the accuracy of the segmentations over the baseline of using only the best seed (top seed + SP).

Qualitative results can be seen in Fig. 5.9d. We observe that the model has comparable qualitative and quantitative performance to the state-of-the-art, despite being a generic model that can readily be applied to other datasets (as opposed to the algorithm of Bo and Fowlkes (2011) which is pedestrian-specific). We consider this possibility in the following section.

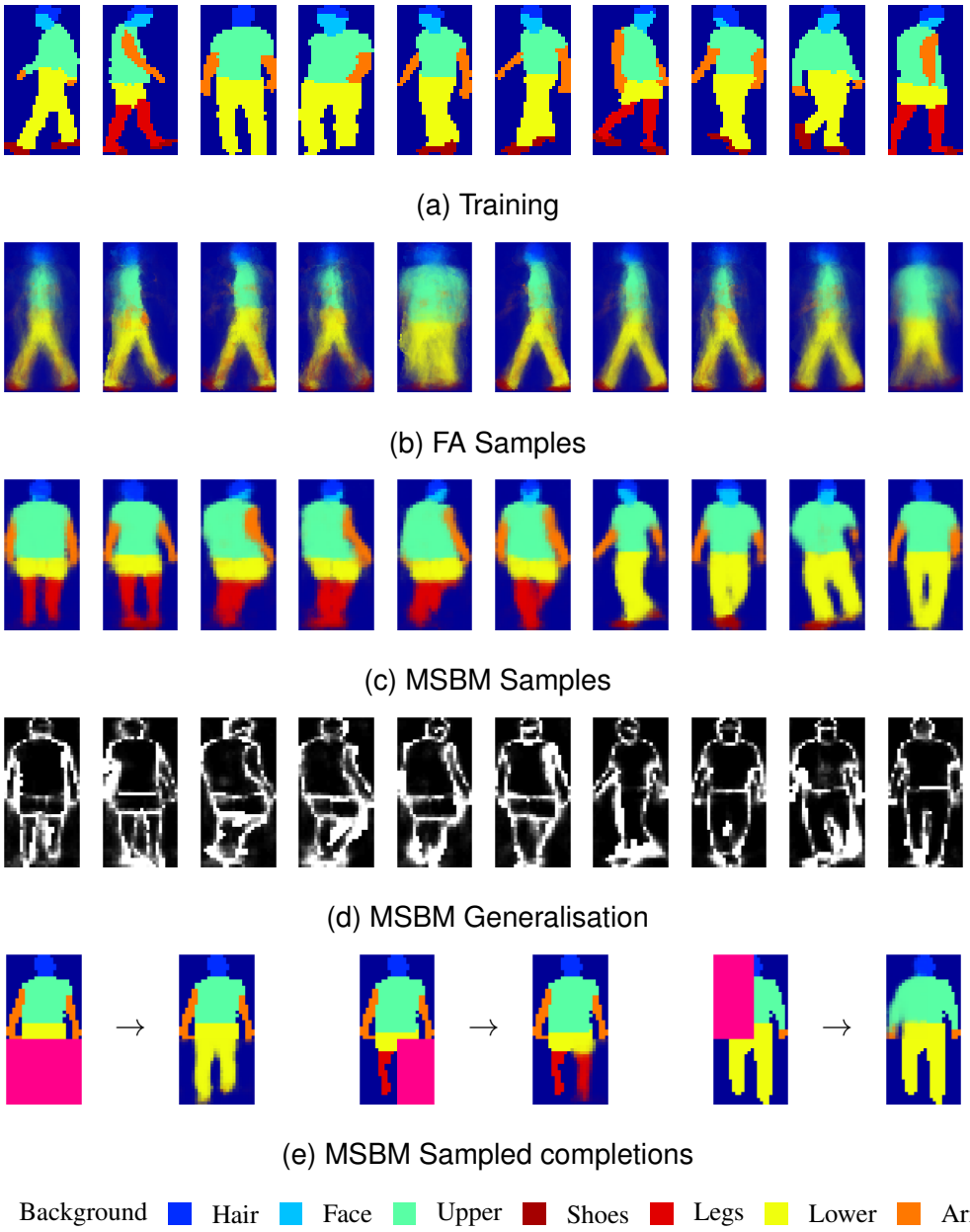


Figure 5.7: **Examining the learned HumanEva dataset shape models**. (a) A selection of images from the dataset. (b) Samples from the learned FA model. (c) A chain of samples from the MSBM (1,000 samples between frames). The apparent 'blurriness' of samples is *not* due to averaging or resizing. We display the *probability* of each pixel belonging to different parts. If, for example, there is a 50-50 chance that a pixel belongs to the red or blue parts, we display that pixel in purple. (d) Differences between the samples and their most similar counterparts in the training dataset. The model generalises in interesting and non-trivial ways to pedestrian shapes not present in the training data. (e) Sampled completions of occlusions (pink). For each occlusion we show an example completion.

5.4.2 ETHZ cars

The second dataset that we considered is the ETHZ labelled cars dataset (Thomas et al., 2009), which itself is a subset of the LabelMe dataset (Russell et al., 2008). It consists of 139 images of cars, all in the same semi-profile view. We used the associated ground-truth segmentations for $L = 6$ parts (body, wheel, window, bumper, license plate, headlight; see Fig. 5.8a for examples). The images are annotated with ground-truth segmentations for $L = 6$ parts (body, wheel, window, bumper, license plate, headlight). We compare the performance of the model with the ISM of Thomas et al. (2009), who also report their results on this dataset.

The dataset was split into 10 train/test cross-validation splits without replacement. We used the training images in each split to train both the shape and appearance components. For the shape component, we trained an SBM at 50×50 pixels with overlap $r = 4$, and 2000 and 100 hidden units in the first and second layers respectively. Each layer was pre-trained for 3000 epochs and joint training was performed for 1000 epochs. We also trained an FA shape model with $H = 2$ latent space dimensions for comparison (see Sec. 5.4.3 for details).

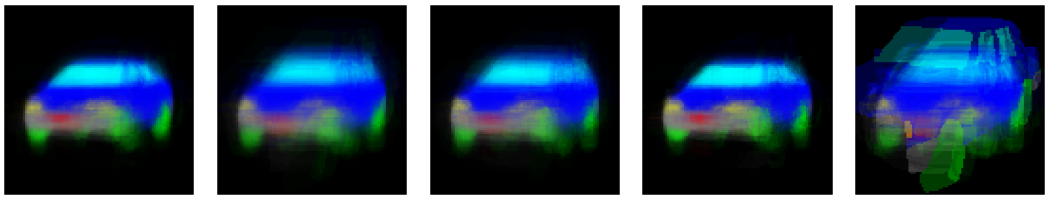
The appearance model was trained with a vocabulary of size $W = 50$ and $K = 100$ mixture components and we set $\lambda = 0.7$. Inference chains were seeded at 50 exemplar segmentations (obtained using K-medoids). We found that the use of superpixels did not help with this dataset (due to the poor quality of superpixels obtained for these images).

Qualitative and quantitative results that show the performance of model to be comparable to the state-of-the-art ISM can be seen in Fig. 5.10b and Table 5.2. The discrepancy in accuracy between the MSBM and ISM on the 'license' and 'light' labels appears to be mainly due to ISM's use of interest-points, as they are able to locate such fine structures accurately. By incorporating better models of part appearance into the generative model, we could expect to see this discrepancy decrease.

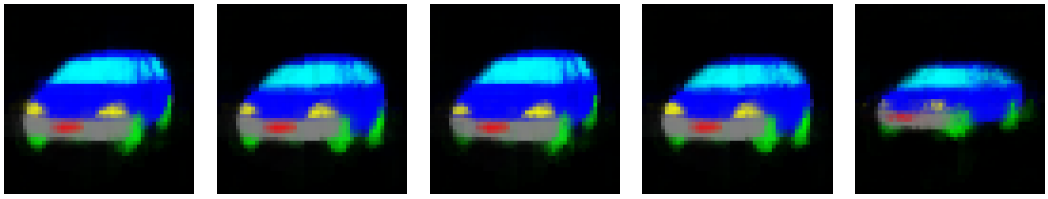
5.4.3 Comparison with the Factor Analysis shape model

In order to be able to perform a comparison, we also trained a Factor Analysis (FA) shape model (i.e. the model presented in Chapter 3) on the Penn-Fudan pedestrian and

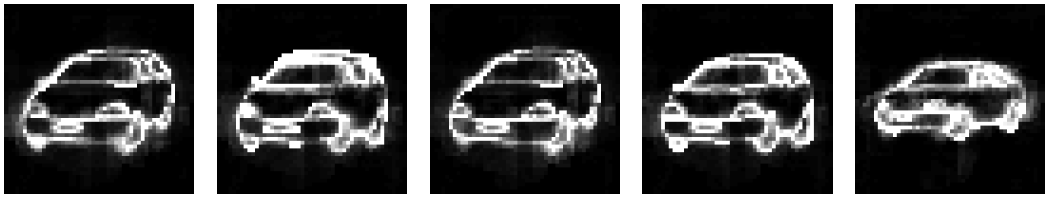
(a) Training



(b) FA Samples



(c) MSBM Samples



(d) MSBM Generalisation

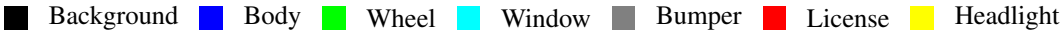


Figure 5.8: **Examining the learned ETHZ cars shape models**. (a) Examples from the training data. Different colours represent different object parts. (b) Samples from the learned FA model. (c) A chain of samples from the MSBM (1,000 samples between frames). The apparent 'blurriness' of samples is *not* due to averaging or resizing. We display the *probability* of each pixel belonging to different parts. If, for example, there is a 50-50 chance that a pixel belongs to the red or blue parts, we display that pixel in purple. (d) Differences between the MSBM samples and their most similar counterparts in the training dataset.

the ETHZ car datasets. The training and testing schedules, as well as the parameters of the appearance model, remained unchanged from Secs. 5.4.1 and 5.4.2.

We trained FA models with $H = 2$, 5 and 10 latent space dimensions on both datasets, and in Tables 5.1 and 5.2 we report results using $H = 5$ and 2 for the Penn-Fudan and ETHZ datasets respectively. These settings of the parameter H were found to obtain the highest scores for the two datasets.

As can be seen in Tables 5.1 and 5.2, FA's performance is less competitive on these more challenging datasets when compared to the MSBM. We note that it may be possible to improve the accuracy of FA segmentations by optimising the parameters of the appearance models that are used in conjunction with the FA shape models.

5.5 Conclusions

In this chapter we have shown how the SBM can be extended to obtain the MSBM, and have presented a principled probabilistic model of images of objects that exploits the MSBM as its model for part shapes. We demonstrated how object segmentations can be obtained simply by performing MCMC inference in the model.

The model can also be treated as a probabilistic *evaluator* of segmentations: given a proposal segmentation it can be used to estimate its likelihood. This leads us to believe that the combination of a generative model such as ours, with a discriminative, bottom-up segmentation algorithm could be highly effective.

Table 5.1: **Results on the Penn-Fudan pedestrians dataset**. We report the percentage of correctly labelled pixels. The first column (FG) is the percentage of correctly identified non-background labels. The final column (Average) is an average of the background, upper and lower body scores (as reported in Bo and Fowlkes, 2011).

	FG	BG	Upper Body	Lower Body	Head	Average
Bo et al. (2011)	**73.3%**	**81.1%**	**73.6%**	**71.6%**	51.8%	**69.5%**
FA	65.4%	66.9%	63.6%	58.6%	49.0%	59.5%
MSBM	70.7%	72.8%	68.6%	66.7%	53.0%	65.3%
MSBM + SP	71.6%	73.8%	69.9%	68.5%	**54.1%**	66.6%
Top seed	59.0%	61.8%	56.8%	49.8%	45.5%	53.5%
Top seed + SP	61.6%	67.3%	60.8%	54.1%	43.5%	56.4%

Table 5.2: **Results on the ETHZ cars dataset**. We report the percentage of pixels belonging to each part that are labelled correctly. The final column is an average weighted by the frequency of occurrence of each label.

	BG	Body	Wheel	Window	Bumper	License	Light	Average
ISM	93.2%	72.2%	**63.6%**	**80.5%**	**73.8%**	**56.2%**	**34.8%**	**86.8%**
FA	87.9%	70.1%	31.0%	70.0%	51.1%	16.6%	25.7%	79.8%
MSBM	**94.6%**	**72.7%**	36.8%	74.4%	64.9%	17.9%	19.9%	86.0%
Top seed	92.2%	68.4%	28.3%	63.8%	45.4%	11.2%	15.1%	81.8%

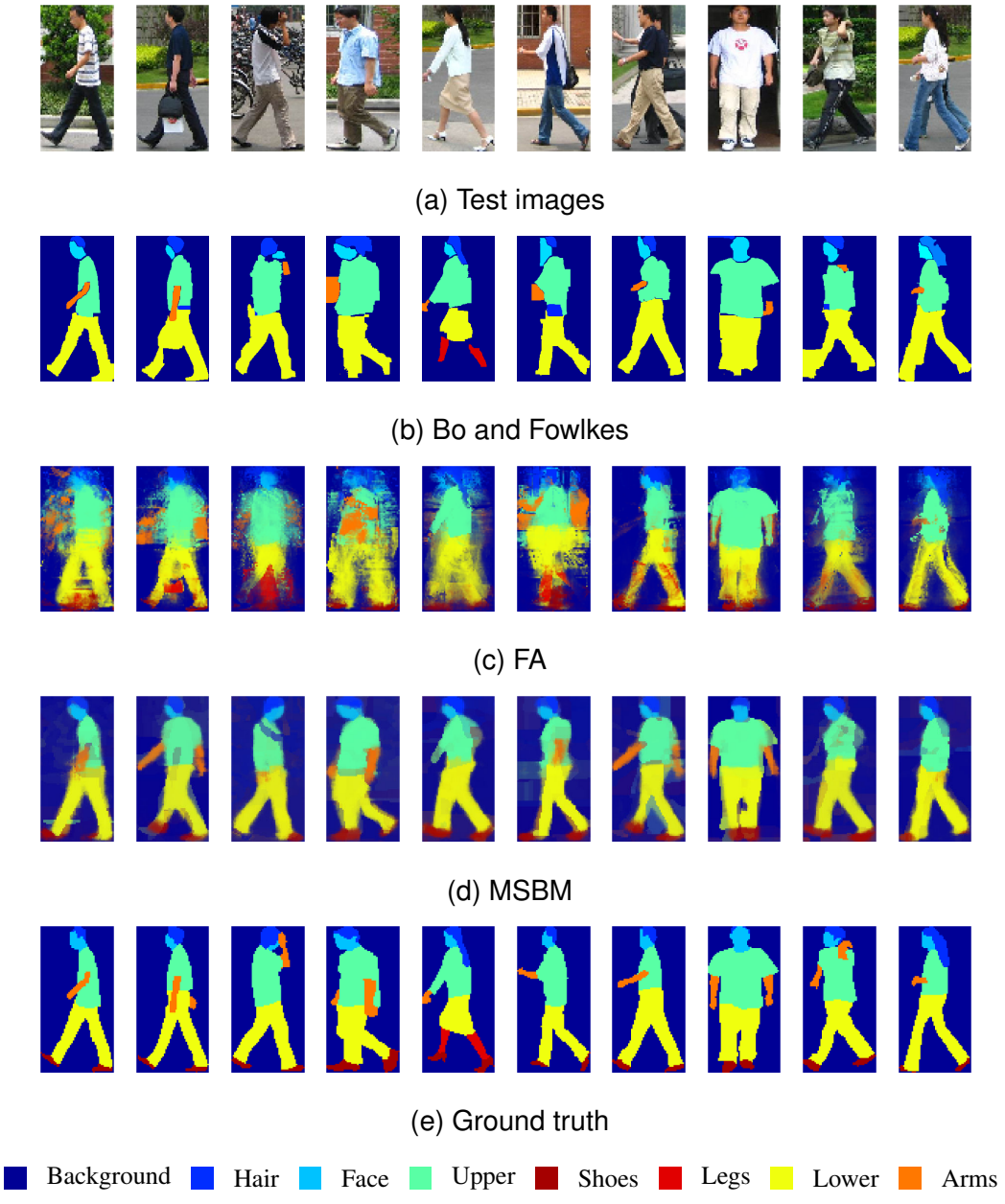


Figure 5.9: **Results on the Penn-Fudan pedestrians dataset**. (a) Test images. (b) Results reported by Bo and Fowlkes (Bo and Fowlkes, 2011). (c) Output of the FA model. (d) Output of the MSBM model. (e) Ground-truth images. The images shown are those selected by Bo and Fowlkes (2011).

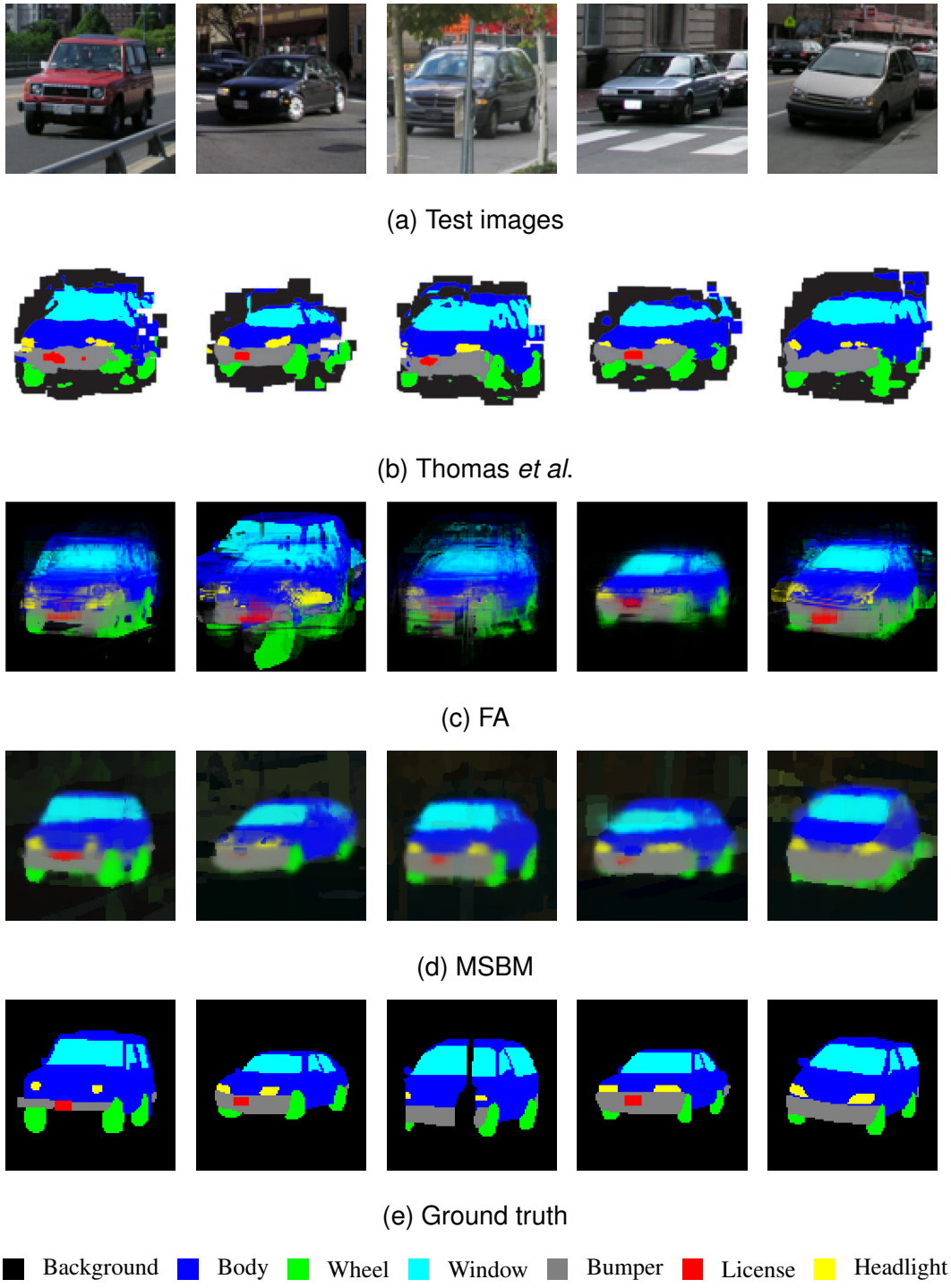


Figure 5.10: **Results on the ETHZ cars dataset**. (a) Test images. (b) Results reported by Thomas et al. (Thomas et al., 2009). (c) Output of the FA model. (d) Output of the MSBM model. (e) Ground-truth images. The images shown are those selected by Thomas et al. (2009).

Chapter 6

Conclusions and Future Work

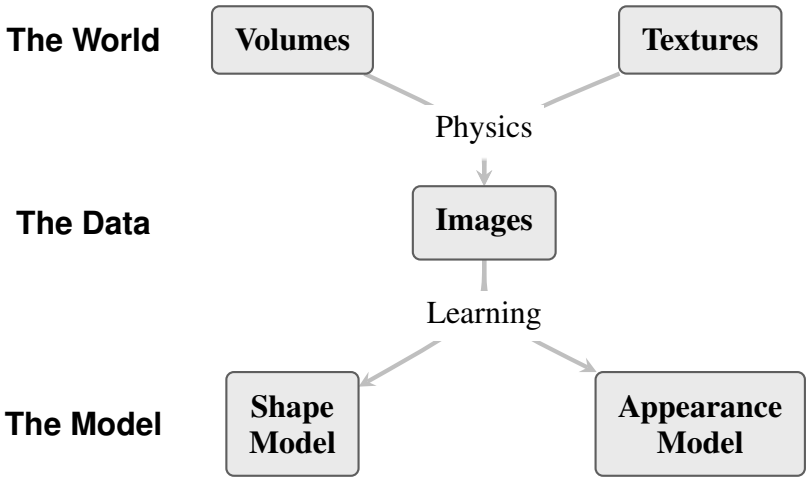


Figure 6.1: Modelling the world through shapes and appearances.

6.1 Summary of the thesis

Parts-based segmentation is challenging primarily due to the huge amount of variability one sees in images of natural scenes. A large number of factors combine in complex ways to generate the pixel intensities that make up any given image. These factors include, but are not limited to, object pose, appearance and shape, camera pose and scene illumination.

When the objects colours are near constant in the dataset, e.g. in videos, statistics of their pixel colours have been used to guide segmentation. However, for many datasets

of interest object appearances are too variable to be modelled with accuracy. In this thesis we focused on developing principled probabilistic models that allow us to incorporate knowledge about shapes for the segmentation task.

6.1.1 The Factored Shapes and Appearances framework

First, in Chapter 3 we presented a novel parts-based image representation that learns from unlabelled images that exhibit variability in both the shapes and appearances of objects. Through experiments on a variety of datasets we demonstrated the advantages of explicitly modelling shape variability. We also showed that the model's latent representations can be interpreted as 'parsings' of images. We applied the model to the object segmentation task, and found that its performance is comparable to that of the state-of-the-art on a number of benchmark datasets.

6.1.2 The Shape Boltzmann Machine

Second, in Chapter 4 we focussed on the task of constructing accurate models of shapes. We presented a type of Deep Boltzmann Machine that we call a *Shape Boltzmann Machine* (SBM) for the task of modelling foreground/background (binary) shapes. We showed that the SBM characterises a 'strong' model of shape, in that samples from the model look realistic and that it generalises to generate samples that differ from training examples. We demonstrated that the SBM learns distributions that are qualitatively and quantitatively better than existing models at this task.

6.1.3 A Boltzmann machine model for parts-based object segmentation

Third, in Chapter 5 we extended the SBM to account for multi-part shapes to obtain the Multinomial Shape Boltzmann Machine (MSBM), and demonstrated how the SBM can be used in conjunction with an appearance model to form a fully generative model of images of objects. We showed how parts-based object segmentations can be obtained simply by performing probabilistic inference in this joint model. We applied the model to several challenging datasets and found its performance to be comparable to the state-of-the-art.

6.2 Discussion

Thanks to their formulations as generative models, the models presented in this thesis are versatile, in that they can be applied to a wide range of data and used for a variety of tasks. However, there are several open questions associated with such applications of these models:

6.2.1 Image resolution

First, our shape models are currently of relatively low resolution compared to many real-world images. Naïvely scaling up the models is unlikely to work as this would greatly increase the number of parameters (and hence the potential to overfit) and also lead to practical problems such as slow mixing when sampling from the model.

In Chapter 5 we demonstrated one approach for side-stepping this problem by upsampling the predictions of the low-resolution shape prior at test-time. This appears to work well in practice but ultimately still limits the level of detail at which shapes can be modelled.

An alternative approach could be to model the activations of a suitable set of fixed basis functions (e.g. fixed Gaussians distributed with some pattern over the image) instead of directly modelling the pixels. However, it is not fully clear how these basis functions should be chosen in order to meaningfully increase the model's ability to represent high resolution details whilst maintaining desirable properties such as spatial locality.

6.2.2 Multiple objects

The second question is how to handle real-world images that contain not just one but many objects. This will make it necessary to model the interactions between the shapes of multiple occluding objects. Although the MSBM can model multiple regions (see Chapter 5) it is unlikely to be a good model of the regions that are the result of occlusion.

As discussed e.g. in Lücke et al., 2009 and Le Roux et al., 2011, it is possible to explicitly reason about object ordering and occlusion when parsing a scene. Such

solutions are, in principle, directly applicable to the SBM and it would be of interest to investigate how they can be utilised together in practical applications.

6.2.3 Aspect variability

The third question is that of aspect variability. Most classes of objects exhibit varying types of outlines when viewed from different angles, and the relationship between these different types is often difficult to model.

One possible approach would be to capture the variability of each view of the object with a separate shape model, and combine the resulting models to create a large *mixture*. To segment any given image, our algorithms would then have to consider the likelihood of the image being generated by each one of these shape models. The downside of this approach is that in naïve implementations information will not transfer across aspects, potentially increasing the amount of data required for training, although this problem can be mitigated to some degree (see e.g. Thomas et al., 2006; cf. 3D models of e.g. Hoiem et al., 2007).

6.2.4 Translation, rotation and scale invariance

We finally highlight the issues of translation, rotation and scale invariance. The models that we described in this thesis work best when the object appears in roughly the same location in every image. This limits their applicability to most collections of images, including modern benchmark datasets such as PASCAL VOC (Everingham et al., 2010) and ImageNet (Deng et al., 2009), which have objects appearing in different locations in the image plane. Although the presented shape models should be able to learn about some of these transformations, they will need to be exposed to training data in all possible positions, rotations and scales in order to do so, and they will not share their parameters across the different locations effectively. These invariances are challenges for many dense, pixel-level models, not just FSA and the SBM.

One approach to this problem is to use convolutional architectures (e.g. Desjardins and Bengio, 2008; Roth and Black, 2005; Ranzato et al., 2010). Such architectures are inherently translation invariant but can be difficult to train and highly computationally expensive.

Extending such convolutional models, Kivinen and Williams (2011) develop translation and rotation 'equivariant' RBMs by augmenting each hidden unit with a latent transformation variable, and show that with this scheme it is possible to learn representations that remain stable in the face of object translation and rotation. This stability can be useful if, for example, the model's latent inferences are used for object classification.

An alternative way to achieve large-scale translation invariance is through a model that is defined only for a tight bounding box enclosing the shape and which is then explicitly translated to all possible image positions via latent transformation variables. Examples of such models include Frey et al. (2003) and Williams and Titsias (2004), and can be considered analogous to their sliding window counterparts in the context of object detection (e.g. Rowley et al., 1998; Schneiderman, 2000; Felzenszwalb et al., 2009).

We believe that by further increasing the number of layers in models such as the SBM, in combination with appropriate constraints on the connectivity we will be able to make progress with respect to both this question and that of resolution (Sec. 6.2.1).

As demonstrated in Sec. 4.4.2.2, when combined with joint training the hierarchical formulation leads to a 'division of labour' across layers, in which the lower layer is responsible for the local details while the higher layer determines the overall pose. This allows the model to *learn* some degree of small-scale invariances, achieving an effect similar to pooling (e.g. as in Lee et al., 2009), but without having to explicitly build it into the model.

We expect that deeper models, in which such effects are replicated across several layers, will be able to handle larger invariances, and will be used to model shapes at higher resolutions.

Appendix A

Inference and Learning for FSA

A.1 Samples of $p(\mathbf{A}, \mathbf{S}, \mathbf{v}|\mathbf{X})$

We derive expressions for the distributions of $p(\mathbf{A}|\mathbf{S}, \mathbf{v}, \mathbf{X}, \boldsymbol{\theta})$ and $p(\mathbf{S}|\mathbf{A}, \mathbf{v}, \mathbf{X}, \boldsymbol{\theta})$ which are used to draw samples for $\mathbf{A}$ and $\mathbf{S}$. Samples for $\mathbf{v}$ are drawn using an elliptical slice sampling procedure. These samples are combined in a block Gibbs MCMC scheme to infer the state of the latent variables given an image $\mathbf{X}$. Observe that

$$p(\mathbf{S}|\mathbf{A}, \mathbf{v}, \mathbf{X}, \boldsymbol{\theta}) = \frac{p(\mathbf{X}, \mathbf{A}, \mathbf{S}, \mathbf{v}|\boldsymbol{\theta})}{p(\mathbf{A}, \mathbf{v}, \mathbf{X}|\boldsymbol{\theta})} = \frac{p(\mathbf{X}, \mathbf{A}, \mathbf{S}, \mathbf{v}|\boldsymbol{\theta})}{\sum_{\mathbf{S}} p(\mathbf{X}, \mathbf{A}, \mathbf{S}, \mathbf{v}|\boldsymbol{\theta})} \tag{A.1}$$

$$p(\mathbf{A}|\mathbf{S}, \mathbf{v}, \mathbf{X}, \boldsymbol{\theta}) = \frac{p(\mathbf{X}, \mathbf{A}, \mathbf{S}, \mathbf{v}|\boldsymbol{\theta})}{p(\mathbf{S}, \mathbf{v}, \mathbf{X}|\boldsymbol{\theta})} = \frac{p(\mathbf{X}, \mathbf{A}, \mathbf{S}, \mathbf{v}|\boldsymbol{\theta})}{\sum_{\mathbf{A}} p(\mathbf{X}, \mathbf{A}, \mathbf{S}, \mathbf{v}|\boldsymbol{\theta})}. \tag{A.2}$$

To calculate $p(\mathbf{S}|\mathbf{A}, \mathbf{v}, \mathbf{X}, \boldsymbol{\theta})$, we first calculate the denominator $\sum_{\mathbf{S}} p(\mathbf{X}, \mathbf{A}, \mathbf{S}, \mathbf{v}|\boldsymbol{\theta})$:

$$\begin{aligned}
&\sum_{\mathbf{S}} p(\mathbf{X}, \mathbf{A}, \mathbf{S}, \mathbf{v}|\boldsymbol{\theta}) \\
&= \sum_{\mathbf{S}} p(\mathbf{v})\, p(\mathbf{A}|\boldsymbol{\theta})\, p(\mathbf{S}|\mathbf{v}, \boldsymbol{\theta})\, p(\mathbf{X}|\mathbf{A}, \mathbf{S}, \boldsymbol{\theta}) && \text{(A.3)} \\
&= p(\mathbf{v})\, p(\mathbf{A}|\boldsymbol{\theta}) \sum_{\mathbf{S}} p(\mathbf{S}|\mathbf{v}, \boldsymbol{\theta})\, p(\mathbf{X}|\mathbf{A}, \mathbf{S}, \boldsymbol{\theta}) && \text{(A.4)} \\
&= p(\mathbf{v})\, p(\mathbf{A}|\boldsymbol{\theta}) \sum_{\mathbf{S}} \prod_{d=1}^{D} p(\mathbf{s}_d|\mathbf{v}, \boldsymbol{\theta})\, p(\mathbf{x}_d|\mathbf{A}, \mathbf{s}_d, \boldsymbol{\theta}) && \text{(A.5)} \\
&= p(\mathbf{v})\, p(\mathbf{A}|\boldsymbol{\theta}) \prod_{d=1}^{D} \sum_{\mathbf{s}_d} p(\mathbf{s}_d|\mathbf{v}, \boldsymbol{\theta})\, p(\mathbf{x}_d|\mathbf{A}, \mathbf{s}_d, \boldsymbol{\theta}). && \text{(A.6)}
\end{aligned}$$

Plugging into Eq. A.1, we get

$$p(\mathbf{S}|\mathbf{A},\mathbf{v},\mathbf{X},\boldsymbol{\theta}) = \frac{p(\mathbf{v})\,p(\mathbf{A}|\boldsymbol{\theta})\prod_{d=1}^{D} p(\mathbf{s}_d|\mathbf{v},\boldsymbol{\theta})\,p(\mathbf{x}_d|\mathbf{A},\mathbf{s}_d,\boldsymbol{\theta})}{p(\mathbf{v})\,p(\mathbf{A}|\boldsymbol{\theta})\prod_{d=1}^{D}\sum_{\mathbf{s}_d} p(\mathbf{s}_d|\mathbf{v},\boldsymbol{\theta})\,p(\mathbf{x}_d|\mathbf{A},\mathbf{s}_d,\boldsymbol{\theta})} \tag{A.7}$$

$$= \prod_{d=1}^{D}\left(\frac{p(\mathbf{s}_d|\mathbf{v},\boldsymbol{\theta})\,p(\mathbf{x}_d|\mathbf{A},\mathbf{s}_d,\boldsymbol{\theta})}{\sum_{\mathbf{s}_d} p(\mathbf{s}_d|\mathbf{v},\boldsymbol{\theta})\,p(\mathbf{x}_d|\mathbf{A},\mathbf{s}_d,\boldsymbol{\theta})}\right). \tag{A.8}$$

Similarly for $p(\mathbf{A}|\mathbf{S},\mathbf{v},\mathbf{X},\boldsymbol{\theta})$, we first calculate the denominator $\sum_{\mathbf{A}} p(\mathbf{X},\mathbf{A},\mathbf{S},\mathbf{v}|\boldsymbol{\theta})$:

$$\sum_{\mathbf{A}} p(\mathbf{X},\mathbf{A},\mathbf{S},\mathbf{v}|\boldsymbol{\theta}) = \sum_{\mathbf{A}} p(\mathbf{v})\,p(\mathbf{A}|\boldsymbol{\theta})\,p(\mathbf{S}|\mathbf{v},\boldsymbol{\theta})\,p(\mathbf{X}|\mathbf{A},\mathbf{S},\boldsymbol{\theta}) \tag{A.9}$$

$$= p(\mathbf{v})\,p(\mathbf{S}|\mathbf{v},\boldsymbol{\theta})\sum_{\mathbf{A}} p(\mathbf{A}|\boldsymbol{\theta})\,p(\mathbf{X}|\mathbf{A},\mathbf{S},\boldsymbol{\theta}). \tag{A.10}$$

Now focussing on the $\sum_{\mathbf{A}} p(\mathbf{A}|\boldsymbol{\theta})\,p(\mathbf{X}|\mathbf{A},\mathbf{S},\boldsymbol{\theta})$ we have

$$\sum_{\mathbf{A}} p(\mathbf{A}|\boldsymbol{\theta})\,p(\mathbf{X}|\mathbf{A},\mathbf{S},\boldsymbol{\theta})$$

$$= \sum_{\mathbf{A}}\left[\left(\prod_{l=0}^{L}\prod_{k=1}^{K}(\pi_{lk})^{a_{lk}}\right)\times\left(\prod_{d=1}^{D}\left(\prod_{l=0}^{L}\left(\prod_{k=1}^{K}\left(\sum_{w=1}^{W}\phi_{lkw}\,\mathcal{N}(\mathbf{x}_d|\boldsymbol{\mu}_w,\boldsymbol{\Sigma}_w)\right)^{a_{lk}}\right)^{s_{ld}}\right)\right)\right]$$

$$= \sum_{\mathbf{A}}\prod_{l=0}^{L}\left[\left(\prod_{k=1}^{K}(\pi_{lk})^{a_{lk}}\right)\times\left(\prod_{d=1}^{D}\prod_{k=1}^{K}\left(\sum_{w=1}^{W}\phi_{lkw}\,\mathcal{N}(\mathbf{x}_d|\boldsymbol{\mu}_w,\boldsymbol{\Sigma}_w)\right)^{a_{lk}\cdot s_{ld}}\right)\right] \tag{A.11}$$

$$= \sum_{\mathbf{A}}\prod_{l=0}^{L}\left[\prod_{k=1}^{K}\left[\pi_{lk}\prod_{d=1}^{D}\left(\sum_{w=1}^{W}\phi_{lkw}\,\mathcal{N}(\mathbf{x}_d|\boldsymbol{\mu}_w,\boldsymbol{\Sigma}_w)\right)^{s_{ld}}\right]^{a_{lk}}\right] \tag{A.12}$$

$$= \prod_{l=0}^{L}\sum_{\mathbf{a}_l}\left[\prod_{k=1}^{K}\left[\pi_{lk}\prod_{d=1}^{D}\left(\sum_{w=1}^{W}\phi_{lkw}\,\mathcal{N}(\mathbf{x}_d|\boldsymbol{\mu}_w,\boldsymbol{\Sigma}_w)\right)^{s_{ld}}\right]^{a_{lk}}\right] \tag{A.13}$$

$$= \prod_{l=0}^{L}\sum_{k=1}^{K}\left[\pi_{lk}\prod_{d=1}^{D}\left(\sum_{w=1}^{W}\phi_{lkw}\,\mathcal{N}(\mathbf{x}_d|\boldsymbol{\mu}_w,\boldsymbol{\Sigma}_w)\right)^{s_{ld}}\right]. \tag{A.14}$$

Plugging into Eq. A.2, we get

$$
\begin{aligned}
&p(\mathbf{A}|\mathbf{S},\mathbf{v},\mathbf{X},\boldsymbol{\theta}) \\
&= \frac{p(\mathbf{v})\,p(\mathbf{S}|\mathbf{v},\boldsymbol{\theta})p(\mathbf{A}|\boldsymbol{\theta})\,p(\mathbf{X}|\mathbf{A},\mathbf{S},\boldsymbol{\theta})}{p(\mathbf{v})\,p(\mathbf{S}|\mathbf{v},\boldsymbol{\theta})\sum_{\mathbf{A}} p(\mathbf{A}|\boldsymbol{\theta})\,p(\mathbf{X}|\mathbf{A},\mathbf{S},\boldsymbol{\theta})} \qquad \text{(A.15)} \\
&= \prod_{l=0}^{L}\left(\frac{\prod_{k=1}^{K}\left[\pi_{lk}\prod_{d=1}^{D}\left(\sum_{w=1}^{W}\phi_{lkw}\,\mathcal{N}(\mathbf{x}_d|\boldsymbol{\mu}_w,\boldsymbol{\Sigma}_w)\right)^{s_{ld}}\right]^{a_{lk}}}{\sum_{k=1}^{K}\left[\pi_{lk}\prod_{d=1}^{D}\left(\sum_{w=1}^{W}\phi_{lkw}\,\mathcal{N}(\mathbf{x}_d|\boldsymbol{\mu}_w,\boldsymbol{\Sigma}_w)\right)^{s_{ld}}\right]}\right). \qquad \text{(A.16)}
\end{aligned}
$$

In other words

$$
p(a_{lk}=1|\mathbf{S},\mathbf{v},\mathbf{X},\boldsymbol{\theta}) = \frac{\pi_{lk}\prod_{d=1}^{D}\left(\sum_{w=1}^{W}\phi_{lkw}\,\mathcal{N}(\mathbf{x}_d|\boldsymbol{\mu}_w,\boldsymbol{\Sigma}_w)\right)^{s_{ld}}}{\sum_{r=1}^{K}\left[\pi_{lr}\prod_{d=1}^{D}\left(\sum_{w=1}^{W}\phi_{lrw}\,\mathcal{N}(\mathbf{x}_d|\boldsymbol{\mu}_w,\boldsymbol{\Sigma}_w)\right)^{s_{ld}}\right]}. \qquad \text{(A.17)}
$$

A.2 The derivative of Q

For the M step of the EM algorithm we wish to find $\arg\max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta},\boldsymbol{\theta}^{\text{old}})$. In this section we derive these updates for $\boldsymbol{\theta}$. By definition, we have

$$
\begin{aligned}
Q(\boldsymbol{\theta},\boldsymbol{\theta}^{\text{old}}) &= \ln p(\boldsymbol{\theta}) + \sum_{i=1}^{n}\sum_{\mathbf{Z}^i} p(\mathbf{Z}^i|\mathbf{X}^i,\boldsymbol{\theta}^{\text{old}})\,\ln p(\mathbf{X}^i,\mathbf{Z}^i|\boldsymbol{\theta}) \qquad \text{(A.18)} \\
&= \ln p(\boldsymbol{\theta}) + \sum_{i=1}^{n}\sum_{\mathbf{Z}^i} p(\mathbf{Z}^i|\mathbf{X}^i,\boldsymbol{\theta}^{\text{old}})\,\ln p(\mathbf{X}^i,\mathbf{A}^i,\mathbf{S}^i,\mathbf{v}^i|\boldsymbol{\theta}). \qquad \text{(A.19)}
\end{aligned}
$$

A.2.1 Updates for θ^s

In order to find $\arg\max_{\boldsymbol{\theta}^m} Q(\boldsymbol{\theta},\boldsymbol{\theta}^{\text{old}})$, we require the derivative of Q with respect to each row $\mathbf{F}_{r(q)}$ and c_{rq}.

Expanding the equations and mapping the multiplications inside the $\ln(\cdot)$ function to

additions, we get

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}})$$
$$= \sum_{i=1}^{n} \sum_{\mathbf{Z}^i} \Big[p(\mathbf{Z}^i|\mathbf{X}^i, \boldsymbol{\theta}^{\text{old}}) \times \left(\sum_{d=1}^{D} \ln \left((1 - L\epsilon) \frac{\prod_{l=0}^{L} \exp\{m_{ld}\}^{s^i_{ld}}}{\sum_{k=0}^{L} \exp\{m_{kd}\}} + \epsilon \right) \right) \Big] + const. \tag{A.20}$$

It can be shown that

$$\frac{\partial Q}{\partial \mathbf{F}_{r(q)}} = \sum_{i=1}^{n} \sum_{\mathbf{Z}^i} \Big[p(\mathbf{Z}^i|\mathbf{X}^i, \boldsymbol{\theta}^{\text{old}}) \times (A \cdot \mathbf{v}_r - B \cdot \mathbf{v}_r) \Big] \tag{A.21}$$

and that

$$\frac{\partial Q}{\partial c_{rq}} = \sum_{i=1}^{n} \sum_{\mathbf{Z}^i} \Big[p(\mathbf{Z}^i|\mathbf{X}^i, \boldsymbol{\theta}^{\text{old}}) \times (A - B) \Big], \tag{A.22}$$

where

$$A = \frac{(1 - L\epsilon) \cdot s^i_{rq} \cdot \exp\{m_{rq}\} + \epsilon \cdot \exp\{m_{rq}\}}{(1 - L\epsilon) \cdot \prod_{l=0}^{L} \exp\{m_{ld}\}^{s^i_{ld}} + \epsilon \cdot \sum_{k=0}^{L} \exp\{m_{kq}\}}, \tag{A.23}$$

and

$$B = \frac{\exp\{m_{rq}\}}{\sum_{k=0}^{L} \exp\{m_{kq}\}}. \tag{A.24}$$

A.2.2 Updates for θ^a

In order to find $\arg\max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}})$, we require the derivative of Q with respect to each π_{mr} and ϕ_{mrq}.

However, since for each layer l

$$\sum_{k=1}^{K} \pi_{lk} = 1, \tag{A.25}$$

and for each layer l and class k

$$\sum_{w=1}^{W} \phi_{lkw} = 1, \tag{A.26}$$

we re-parametrise the problem to make it compatible with standard numerical optimisation packages. Specifically, each π_{lk} becomes

$$\pi_{lk} = \frac{\exp\{\alpha_{lk}\}}{\sum_{c=1}^{K} \exp\{\alpha_{lc}\}}, \tag{A.27}$$

and each ϕ_{lkw} becomes

$$\phi_{lkw} = \frac{\exp\{\beta_{lkw}\}}{\sum_{v=1}^{W} \exp\{\beta_{lkv}\}}, \tag{A.28}$$

where the α and β are real-valued continuous variables.

It can be shown that the derivative of

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}}) =$$
$$\ln p(\boldsymbol{\phi}) + \sum_{i=1}^{n} \sum_{\mathbf{Z}^i} \left[p(\mathbf{Z}^i|\mathbf{X}^i, \boldsymbol{\theta}^{\text{old}}) \times (\ln p(\mathbf{A}|\boldsymbol{\theta}) + \ln p(\mathbf{X}|\mathbf{A}, \mathbf{S}, \boldsymbol{\theta})) \right] + const.$$

with respect to α_{mr} is:

$$\frac{\partial Q}{\partial \alpha_{mr}} = \sum_{i=1}^{n} \sum_{\mathbf{Z}^i} \left[p(\mathbf{Z}^i|\mathbf{X}^i, \boldsymbol{\theta}^{\text{old}}) \times \left(a_{mr}^i - \sum_{k=1}^{K} a_{mr}^i \cdot \frac{\exp\{\alpha_{mr}\}}{\sum_{c=1}^{K} \exp\{\alpha_{mc}\}} \right) \right] \tag{A.29}$$

and with respect to β_{mrq} is:

$$\begin{aligned}
&\frac{\partial Q}{\partial \beta_{mrq}} = \\
&\lambda_{\text{self}} \cdot \left(\phi_{mrq} - \phi_{mrq}^2\right) \cdot \left(\ln\left(\phi_{mrq}\right) + 1\right) - \\
&\lambda_{\text{self}} \cdot \sum_{v \neq q} \left(\phi_{mrq} \cdot \phi_{mrv}\right) \cdot \left(\ln\left(\phi_{mrv}\right) + 1\right) + \\
&\lambda_{\text{others}} \cdot \sum_{l \neq m} \left(\phi_{mrq} - \phi_{mrq}^2\right) \cdot \left(\frac{\phi_{mq}}{\phi_{lq}} + \ln\left(\frac{\phi_{mq}}{\phi_{lq}}\right) + 1 \right) / K - \\
&\lambda_{\text{others}} \cdot \sum_{l \neq m} \sum_{v \neq q} \left(\phi_{mrq} \cdot \phi_{mrv}\right) \cdot \left(\frac{\phi_{mv}}{\phi_{lv}} + \ln\left(\frac{\phi_{mv}}{\phi_{lv}}\right) + 1 \right) / K + \\
&\sum_{i=1}^{n} \sum_{\mathbf{Z}^i} \left[p(\mathbf{Z}^i|\mathbf{X}^i, \boldsymbol{\theta}^{\text{old}}) \times \right. \\
&\left. \left(a_{mr}^i \sum_{d=1}^{D} s_{md}^i \cdot \frac{\sum_{u \neq q} \phi_{mru} \cdot \left(\mathcal{N}(\mathbf{x}_d|\boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q) - \mathcal{N}(\mathbf{x}_d|\boldsymbol{\mu}_u, \boldsymbol{\Sigma}_u)\right)}{\sum_{w=1}^{W} \phi_{mrw} \cdot \mathcal{N}(\mathbf{x}_d|\boldsymbol{\mu}_w, \boldsymbol{\Sigma}_w)} \right) \right].
\end{aligned} \tag{A.30}$$

Appendix B

Fine-grained Classification with FSA

In this appendix we present preliminary results of applying the FSA model (Chapter 3) to the fine-grained visual categorisation task, in which the goal is to discriminate between *types* of objects such as motorbikes, or between *subspecies* of animals or plants. Our initial results look promising and we believe this to be a potentially fruitful avenue for future research.

B.1 Fine-grained visual classification

In the fine-grained classification setting the visual distinctions between the categories is subtle, and commonly used features can be sub-optimal representations for discrimination between the different classes. We investigate a continuous treatment of this class of problems, where FSA is used to learn a generative model of all the categories simultaneously. FSA is parts-based, and it learns to *smoothly* morph parts from one type to another.

Inferences in the model correspond to low-dimensional representations of the parts' shapes and appearances. We demonstrate that simple classification methods based on these inferences can outperform existing techniques at the fine-grained categorisation task. In order to evaluate our model we introduce an updated version of the Caltech4 motorbikes dataset (Fei-Fei et al., 2004), where each image has been labelled as belonging to one of 6 different sub-categories.

In Sec. B.2 that follows, we describe how inferences in FSA can be used for this task by considering a synthetic dataset of images. In Sec. B.3 we present our updated version of the Caltech4 dataset (FGM6) and evaluate the method's performance on this dataset. We conclude with a discussion of our results in Sec. B.4.

B.2 Experiments on synthetic data

We first illustrate FSA's application to the fine-grained visual categorisation task with a synthetic dataset of images (Fig. B.1). The images are of an 'object' that is composed of two parts: a thick bar, which is always occluded by a thin bar. The size of the parts vary independently from image to image, as do their colours. The background appears with the same colour throughout the dataset. We assume that there are four underlying classes of object within the dataset. The first class of objects is one where the two parts both appear in a horizontal configuration. The other three classes correspond to the remaining possible permutations of horizontal and vertical parts. See Fig. B.1 for samples of images from each class.

The goal is to classify an image of this kind into one of the four categories at test-time. To do this, we begin by constructing an FSA model of the entire dataset. We train FSA with $L = 2$ parts, $H = 2$ latent dimensions, $K = 3$ appearance classes and a vocabulary size of $W = 30$. Although the connected regions *in each image* can easily be found using appearance cues alone, knowledge about shape is required to assign the connected regions to parts that are semantically stable *across the dataset*.

We train on 100 images of the kind seen in Fig. B.1. We additionally provide ground-truth value of the object's segmentation for 4 of the training images – one per class. This information helps guide the Expectation Maximisation training algorithm in the right direction during the first few iterations.

The model's internal representation of the object can be inspected in a number of different ways. In Fig. B.2a we plot random sample segmentations drawn from the distribution defined by the trained FSA model. The samples exhibit part variability in a way that matches the training data. In Fig. B.2b we plot Hinton diagrams of the columns of the model's learnt $\mathbf{F}_l$ matrices. We see that the columns capture the correct thickness of the parts. The weights in the centre of the images have small magnitudes, as the parts' shapes in that region are captured by the mean masks $\mathbf{c}_l$ instead. Since in this

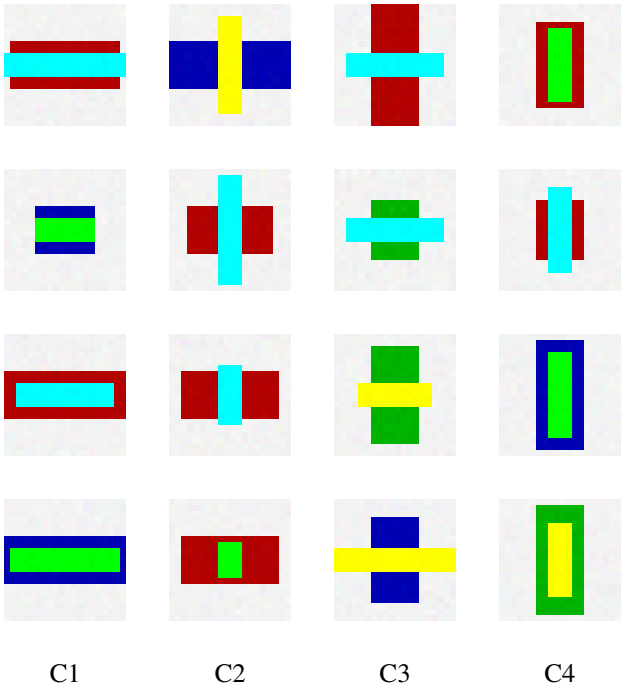


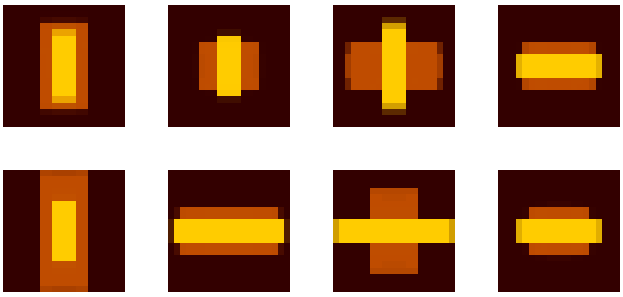
Figure B.1: **Synthetic training data**. A subset of the training images. The classes correspond to the different combinations of vertical and horizontal bars.

example we have chosen to use a 2-dimensional latent space for $\mathbf{v}$ (i.e. $H = 2$), we can conveniently explore this space on a flat plane. In Fig. B.2c we plot the way in which the segmentation generated by the learnt model varies as $\mathbf{v}$ moves in this space.

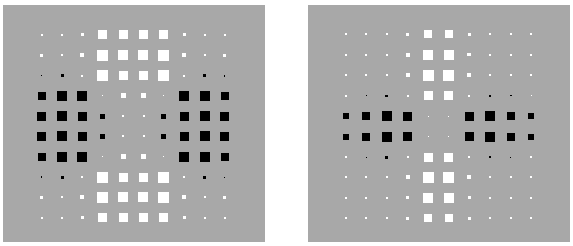
In order to perform classification, we represent each image using a number of 'features'. We can then use any appropriate classification algorithm to learn an accurate mapping from feature-space to category labels. With FSA, we regard the model's inferred representations of the shapes and appearances of the parts as features of the image the object appears in. Specifically, the feature vector $\mathbf{f}^i$ for image $\mathbf{X}^i$ is the $(H + (L + 1) \times K)$-dimensional vector $(\mathbf{v}^i\ \mathbf{a}_0^i\ \mathbf{a}_1^i\ ...\ \mathbf{a}_L^i)^T$.

Fig. B.2c displays the mapping from latent shape-space to category labels learnt by a classification method (details below). The region in which the inferred $\mathbf{v}^i$ for image $\mathbf{X}^i$ falls in this space determines the class that will be assigned to that image. We note that with FSA, the appearances of the parts can also be used as cues for classification.

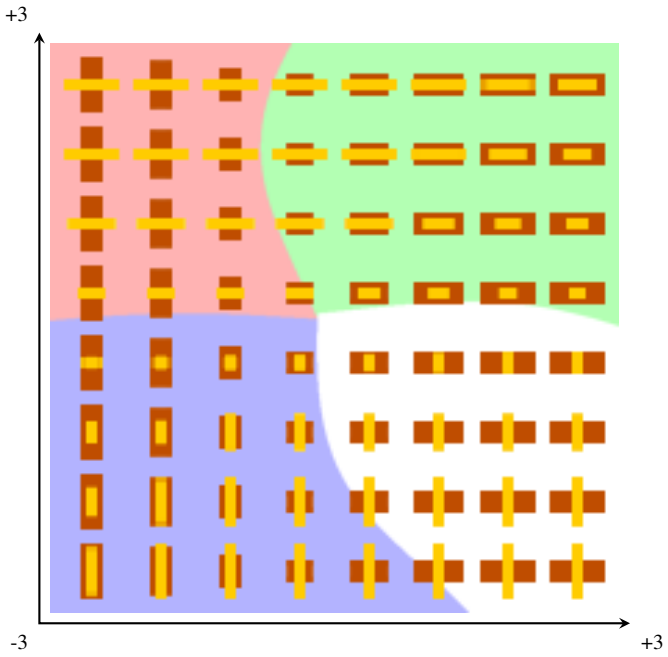
There are a number of different classes of algorithms that can be used for classification. In this evaluation we consider the following three:



(a)



(b)



(c)

Figure B.2: **Inspecting the learned shape model**. (a) Random samples from the FSA model. (b) Hinton diagrams of the columns of $\mathbf{F}_1$ and $\mathbf{F}_2$. (c) A plot of the joint segmentations for a grid of $\mathbf{v}$ values in 2D latent space. The decision boundaries learned by the SVM have been superimposed onto this space.

- **K-nearest neighbours (KNN):** The k-nearest neighbour algorithm is amongst the simplest of all machine learning algorithms. A data vector $\mathbf{f}$ is classified by a majority vote, with its discrete class label $y(\mathbf{f})$ being the class most common amongst its k nearest neighbours. In our experiments we find a choice of $k = 3$ to work well.

- **Generalised linear models (GLM):** This relatively simple family of models can be seen as an extension of the linear regression model. In the GLM the discrete class label $y(\mathbf{f})$ for a data vector $\mathbf{f}$ is given by $f(\mathbf{w}^T\mathbf{f} + w_0)$, where $f(\cdot)$ is commonly known as the *activation function* and $\mathbf{w}$ and w_0 are parameters of the model. We use a softmax activation function in order to perform multi-class classification.

- **Support vector machines (SVM):** A widely used method for classification and regression, SVMs work by learning sparse maximum-margin classifiers in a high dimensional space defined implicitly by a kernel. The idea is that such a classifier will correspond to an accurate non-linear classifier in the original data-space. An important property of SVMs is that the determination of the model parameters corresponds to a convex optimisation problem, and so any local solution is also a global optimum.

See Bishop (2006) for detailed descriptions of these algorithms. We use the NETLAB library (Nabney, 2002) for KNN and GLM classification, and LIBSVM (Chang and Lin, 2011) for SVM classification. The C-SVC SVM is constructed with a radial basis function kernel of the form $k(\mathbf{f}_i, \mathbf{f}_j) = \exp(-\gamma \cdot |\mathbf{f}_i - \mathbf{f}_j|^2)$[1]. We set γ to equal 0.5, as this setting leads to the highest classification performance on a held-out validation dataset.

It is important to note that these methods are sensitive to the scaling of data-dimensions, which is why we normalise the features of the training data to ensure that the range of each dimension lies in $[-1, 1]$.

We train the classifiers by providing the ground-truth labelling of the 100 training images' classes. We test on 500 unlabelled images, achieving average accuracies of 98.0% (KNN), 95.2% (GLM) and 96.6% (SVM). Mislabelled images are typically those where the length of both parts is small.

[1] This is the default kernel provided by the LIBSVM library.

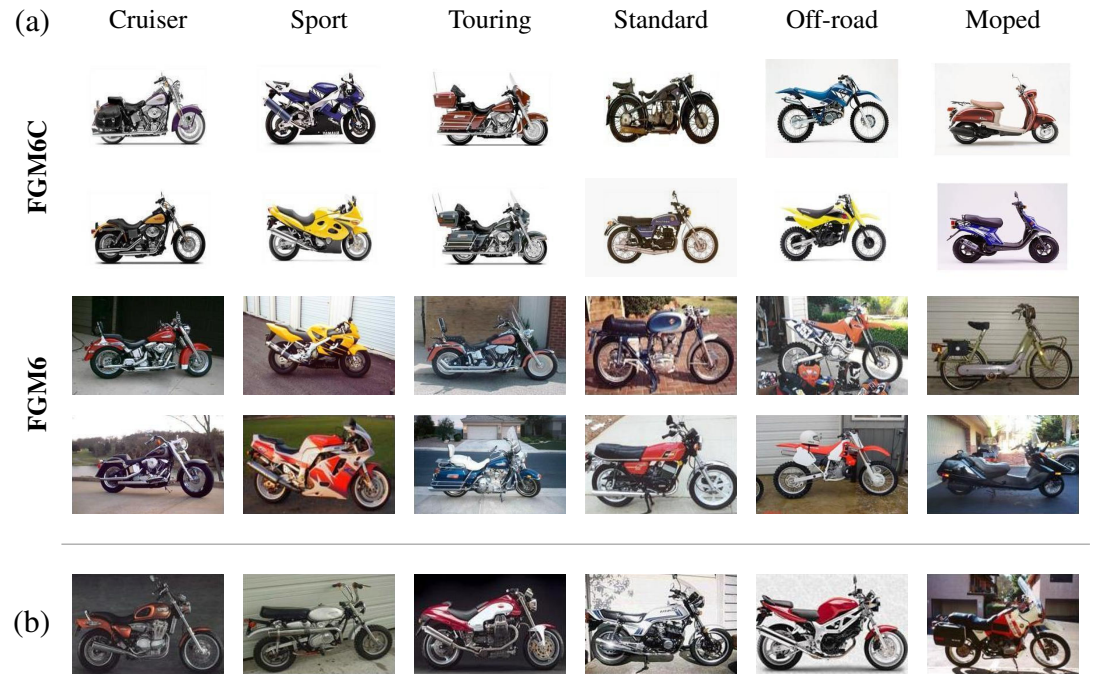


Figure B.3: **FGM6 data**. (a) A selection of images of each class from the FGM6 and FGM6C datasets. (b) A selection of images from the Caltech4 dataset that were not included in the FGM6 dataset, as there was low agreement between the annotators on the class of the motorbikes in the images.

B.3 Experiments on real data

In order to further evaluate the model we run experiments on a real classification task. For this purpose we constructed a 6-class, labelled version of the Caltech4 dataset Fei-Fei et al. (2004), which we refer to as *Fine-Grained Motorbikes 6* (FGM6).

In the fine-grained categorisation setting, the ground-truth value of the label for each image can often be ambiguous. We overcome this ambiguity by using the consensus of the labellings provided by a number of different annotators.

We asked five participants to categorise all 798 original Caltech motorbike images into 6 different classes. These classes include 'Cruiser', 'Sport', 'Touring', 'Standard', 'Moped' and 'Off-road', and were defined by their descriptions on Wikipedia[2]. We

[2] http://en.wikipedia.org/wiki/Types_of_motorcycles. At the time of writing, this page lists 8 categories of motorbikes. However, we chose to merge the 'Sports-touring' and 'Touring' categories, as well as the 'Dual-sport' and 'Off-road' categories due to the rarity of images of those types in the original Caltech dataset.

Table B.1: **FGM6 statistics**. The first two rows correspond to the average 1-versus-rest accuracies between the 5 annotators and the consensus labelling for the two datasets.

		Cruiser	Sport	Touring	Standard	Off-road	Moped	Combined
Annotator Agreement	**FGM6C**	96.5%	96.9%	99.5%	96.7%	97.4%	100.0%	93.5%
	FGM6	95.3%	95.6%	98.6%	95.0%	96.5%	100.0%	90.4%
Size	**FGM6C**	31	54	5	22	42	5	159
	FGM6	158	171	39	101	104	7	580

retained images on which at least four of the five participants had agreed on the labelling, and we discarded the rest. We also consider a 'clean' subset of FGM6, which we refer to as FGM6C, where we only keep the images with uncluttered backgrounds. Examples of images of the six classes from both datasets can be seen in Fig. B.3. In Table B.1 we display further statistics of the datasets. In the experiments that follow, we train on half of the available images and test on the other half.

We consider three baselines algorithms for comparison: Factor Analysis (FA), SIFT features (Lowe, 2004) extracted at interest points, and dense pyramid histogram of gradients features (PHOG, Bosch et al., 2007):

- **FA:** We train a Factor Analysis model of all the training images and use the latent representation under the FA model as the feature vector for each image. We set the number of latent dimensions in the Gaussian prior to equal 20, with which we obtain best results. As such, the feature vector $\mathbf{f}$ is 20-dimensional.
- **SIFT:** We first extract interest points using the Harris-affine keypoint detector (Mikolajczyk and Schmid, 2004). SIFT features are then calculated for each of the keypoints and for all images in the training dataset. The features are clustered using the k-means algorithm to create a so-called *codebook*. Finally, we represent each image with a normalised histogram of occurrences of the codebook features in that image. The idea is that images of a given class are likely to have similar histograms over the codebook features. We obtain best results with $k = 10$. In this case, the feature vector $\mathbf{f}$ is 10-dimensional.
- **PHOG:** Pyramid histogram of gradients is a descriptor that represents local im-

age shape and its spatial layout, together with a spatial pyramid kernel (see Mikolajczyk and Schmid (2004) for further details). The two adjustable parameters of the descriptor are L the number of levels in the pyramid, and B the number of bins in each level. We obtain best results with $L = 3$ and $B = 10$. In this case, the feature vector $\mathbf{f}$ is 1,300-dimensional.

All four methods (FA, SIFT, PHOG and FSA) result in a feature vector $\mathbf{f}$ for each test and training image. We find that normalising these feature vectors results in a significant boost in classification performance. The features are normalised to ensure that each element f_d of the features of the *training* images lies in the range $[-1, 1]$.

To demonstrate that SIFT and PHOG can be powerful tools in the standard classification setting, we use them to classify a dataset of images of motorbikes and horses. The dataset consists of 100 images of motorbikes from the Caltech4 dataset (Fei-Fei et al., 2004) and 100 images of horses from the Weizmann dataset (Borenstein et al., 2004). We report classification accuracies of 91.8% with SIFT and 89.8% with PHOG. Both methods were trained on half of the available data and tested on the other half.

For the FGM6C dataset, we consider two variants of the FSA model. The first variant (referred to as simply FSA) is constructed with $L = 2$ parts, $H = 20$ latent dimensions, $K = 10$ appearance classes and a vocabulary of size $W = 30$. The second variant, which we will refer to as *supervised* FSA or SFSA, is constructed with the same parameters except the vocabulary size is reduced to $W = 20$, and it is provided with human-annotated part segmentations for 18 of the training images. The chosen parts do not have any particular semantic meaning or functional significance, instead they correspond to regions of the bikes whose appearances vary in a relatively well-defined way across the dataset. Examples of these annotations can be seen in Fig. B.4. This helps FSA converge to a parts-based model that matches human intuitions about the nature of parts more closely.

The results of our experiments on the FGM6C dataset can be seen in Table B.2. Supervised FSA combined with an SVM classifier achieves the highest overall classification accuracy (91.9%), correctly labelling 68 of the 74 test images. We note that SFSA's accuracy is close to the average human annotator's accuracy (93.5%). SFSA's performance on 1-versus-rest classification of each motorbike type also beats the other methods for 5 of the 6 categories. See Fig. B.5 for SFSA + SVM's confusion matrix on this dataset.

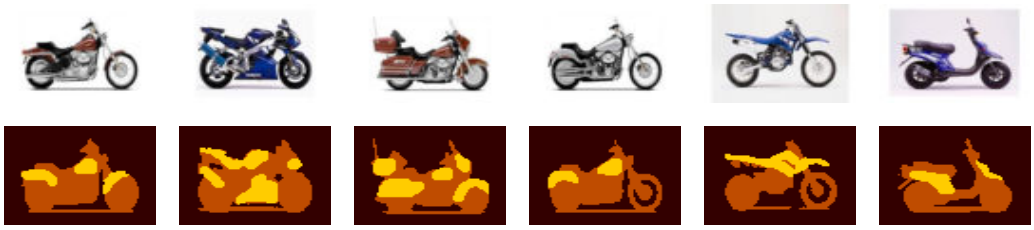


Figure B.4: **A selection of the 18 ground-truth, annotated training images**. Distinct colours indicate assignments of pixels to different parts.

Table B.2: **Results on the FGM6C dataset**. For FA, SIFT and PHOG we report best classification results obtained with any of the KNN, GLM or SVM methods.

		Cruiser	Sport	Touring	Standard	Off-road	Moped	Combined
FA		81.1%	78.4%	97.3%	89.2%	75.7%	97.3%	59.5%
SIFT		86.5%	74.3%	97.3%	78.4%	85.1%	97.3%	59.5%
PHOG		87.8%	**98.6%**	97.3%	89.2%	89.2%	97.3%	79.7%
FSA +	**KNN**	85.1%	93.2%	97.3%	87.8%	95.9%	97.3%	78.4%
	GLM	81.1%	97.3%	97.3%	90.5%	93.2%	97.3%	78.4%
	SVM	90.5%	94.6%	97.3%	89.2%	**98.6%**	97.3%	83.8%
SFSA +	**KNN**	91.9%	93.2%	97.3%	90.5%	91.9%	94.6%	79.7%
	GLM	90.5%	93.2%	97.3%	90.5%	95.9%	94.6%	81.1%
	SVM	**97.3%**	94.6%	**100.0%**	**94.6%**	**98.6%**	**98.6%**	**91.9%**

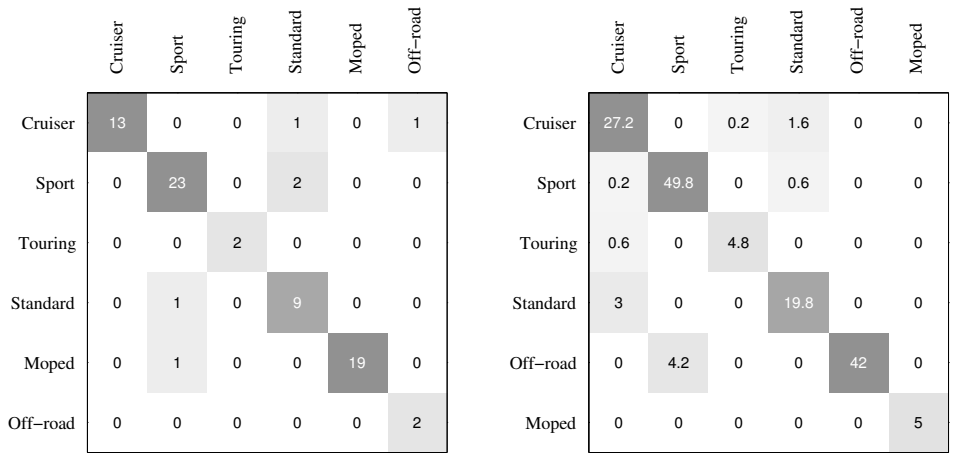


Figure B.5: **FGM6C results**. (a) SFSA+SVM confusion matrix. (b) Average confusion matrix of human annotations.

For the FGM6 dataset, we also consider two variants of the FSA model. The first variant is constructed with $L = 3$ parts, $H = 50$ latent dimensions, $K = 10$ appearance classes and a vocabulary of size $W = 20$. The second variant is constructed with the same parameters except the latent space is reduced to $H = 30$ dimensions, and it is provided with human-annotated segmentations for 18 of the training images.

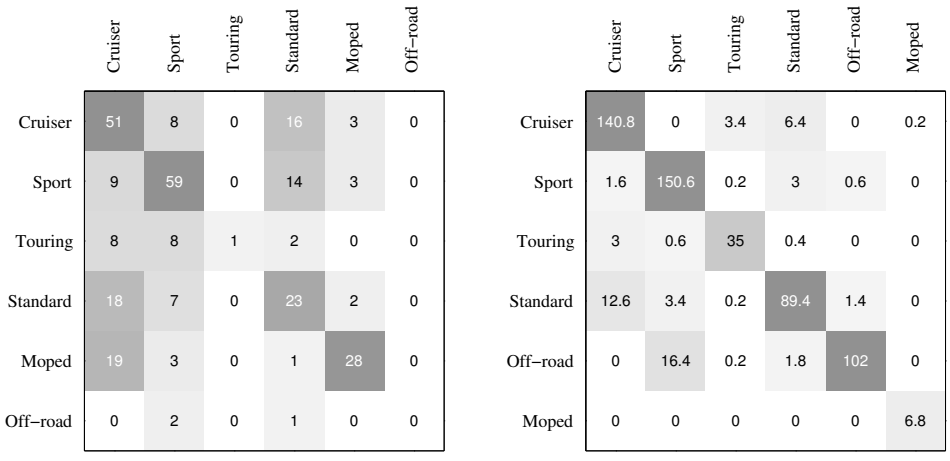
The results of our experiments on the FGM6 dataset can be seen in Table B.3. Supervised FSA combined with an SVM classifier achieves the highest overall classification accuracy (56.6%), correctly labelling 162 of the 286 test images. PHOG accuracy is marginally lower at 55.2%. See Fig. B.6 for SFSA's confusion matrix on this dataset.

The accuracy of all five methods on FGM6 is worse than on FGM6C. For SIFT and PHOG, this discrepancy is largely due to noise introduced into the histogram models by background clutter. FSA's performance drops when it fails to accurately segment the foreground object from the background.

Finally, we re-iterate that FSA features tend to be much lower-dimensional than those generated by PHOG (for FGM6, FSA features have 34 dimensions versus 1,300 dimensions for PHOG), yet their efficacy at the classification task is comparable.

Table B.3: **Results on the FGM6 dataset**. For FA, SIFT and PHOG we report best classification results obtained with any of the KNN, GLM or SVM methods.

		Cruiser	Sport	Touring	Standard	Off-road	Moped	Combined
FA		67.8%	59.4%	91.3%	73.4%	72.4%	98.6%	31.5%
SIFT		64.3%	64.3%	88.5%	73.8%	79.7%	96.5%	33.6%
PHOG		58.0%	**86.7%**	93.4%	**82.5%**	**90.9%**	99.0%	55.2%
FSA +	**KNN**	73.8%	75.2%	83.6%	74.8%	87.1%	97.2%	45.8%
	GLM	28.0%	70.3%	92.0%	**82.5%**	82.2%	**99.0%**	26.9%
	SVM	**74.8%**	70.3%	93.4%	76.2%	88.5%	**99.0%**	51.0%
SFSA +	**KNN**	65.4%	71.0%	90.9%	73.1%	79.4%	97.9%	38.8%
	GLM	68.2%	81.1%	93.0%	78.0%	86.7%	97.9%	52.4%
	SVM	71.7%	81.1%	**93.7%**	78.7%	89.2%	**99.0%**	**56.6%**



(a) (b)

Figure B.6: **FGM6 results**. (a) SFSA+SVM confusion matrix. (b) Average confusion matrix of human annotations.

B.4 Conclusions

In this section we considered the application of FSA to the fine-grained visual categorisation task. Our experiments seem to suggest that FSA can outperform commonly used classification techniques at this task using only interpretable, low-dimensional feature vectors.

FSA is suited to this setting as its approach is inherently *continuous*. We also note that the application of FSA produces a dense segmentation in addition to a classification for each image. Such segmentations can be of use in many other settings too, e.g. when we wish to localise the object, or when we wish to identify the appearance properties of its parts.

Bibliography

Ackley, D., Hinton, G., and Sejnowski, T. (1985). A learning algorithm for boltzmann machines. *Cognitive Science*, 9(1):147–169. 59

Alexe, B., Deselaers, T., and Ferrari, V. (2010). ClassCut for unsupervised class segmentation. In *European Conference on Computer vision*, pages 380–393. 26, 45, 47, 48, 92

Anguelov, D., Srinivasan, P., Koller, D., Thrun, S., Rodgers, J., and Davis, J. (2005). SCAPE: shape completion and animation of people. *ACM Transactions on Graphics (SIGGRAPH) 2005*, 24(3):408–416. 48

Arbelaez, P., Maire, M., Fowlkes, C. C., and Malik, J. (2009). From Contours to Regions: An Empirical Evaluation. In *IEEE Conference on Computer Vision and Pattern Recognition 2009*. 95

Arora, H., Loeff, N., Forsyth, D., and Ahuja, N. (2007). Unsupervised Segmentation of Objects using Efficient Learning. *IEEE Conference on Computer Vision and Pattern Recognition 2007*, pages 1–7. 47, 92

Aujoulat, N. (2005). A horse from the "Panel of the Chinese Horses" from the Lascaux Cave. Ministre de la Culture et de la Communication, France. Available at http://www.york.ac.uk/news-and-events/news/2011/research/cave-paintings/images/.

Bahl, L. R., Bakis, R., Jelinek, F., and Mercer, R. L. (1980). Language-Model / Acoustic-Channel-Model Balance Mechanism. *IBM Technical Disclosure Bulletin No. 7B*, 23:3464–3465. 89

Bertozzi, A., Esedoglu, S., and Gillette, A. (2007). Inpainting of Binary Images Using the Cahn-Hilliard Equation. *IEEE Transactions on Image Processing*, 16(1):285 –291. 48

Biederman, I. (1987). Recognition-by-components: A theory of human image understanding. *Psychological Review*, 94:115–147. 7, 17, 19

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc. 12, 117

Bo, Y. and Fowlkes, C. (2011). Shape-based Pedestrian Parsing. In *IEEE Conference on Computer Vision and Pattern Recognition 2011*. 85, 92, 93, 95, 100, 101

Borenstein, E., Sharon, E., and Ullman, S. (2004). Combining Top-Down and Bottom-Up Segmentation. In *CVPR Workshop on Perceptual Organization in Computer Vision*. 21, 26, 45, 47, 48, 50, 52, 62, 92, 120

Bosch, A., Zisserman, A., and Munoz, X. (2007). Representing Shape with a Spatial Pyramid Kernel. In *Proceedings of the ACM International Conference on Image and Video Retrieval*. 119

Boykov, Y. and Jolly, M.-P. (2001). Interactive Graph Cuts for Optimal Boundary & Region Segmentation of Objects in N-D images. In *International Conference on Computer Vision 2001*, pages 105–112. 3, 13

Brand, M. (1999). An entropic estimator for structure discovery. In *Advances in Neural Information Processing Systems 11*, pages 723–729. 31

Bridle, J. S. (1990). Training Stochastic Model Recognition Algorithms as Networks can lead to Maximum Mutual Information Estimation of Parameters. In *Advances in Neural Information Processing Systems 2*, pages 211–217. 87

Cashman, T. and Fitzgibbon, A. (2012). What Shape are Dolphins? Building 3D Morphable Models from 2D Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (99):1. 5

Cemgil, T., Zajdel, W., and Krose, B. (2005). A Hybrid Graphical Model for Robust Feature Extraction from Video. In *IEEE Conference on Computer Vision and Pattern Recognition 2005*, pages 1158–1165. 3, 50, 52, 63

Chan, T. F. and Shen, J. (2001). Nontexture Inpainting by Curvature-Driven Diffusions. *Journal of Visual Communication and Image Representation*, 12(4):436–449. 48

Chang, C.-C. and Lin, C.-J. (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`. 117

CIE (1978). *Recommendations on Uniform Color Spaces, Color-difference Equations, Psychometric Color Terms*. Bureau Central de la CIE. 31

Cootes, T., Edwards, G., and Taylor, C. (2001). Active Appearance Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23:681–685. 3, 9

Cootes, T., Taylor, C., Cooper, D. H., and Graham, J. (1995). Active shape models—their training and application. *Computer Vision and Image Understanding*, 61:38–59. vii, 17, 18, 50

Crevier, D. (1993). *AI: The Tumultuous History of the Search for Artificial Intelligence*. Basic Books. 1

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). ImageNet: A Large-Scale Hierarchical Image Database. In *IEEE Conference on Computer Vision and Pattern Recognition 2009*. 106

Desjardins, G. and Bengio, Y. (2008). Empirical Evaluation of Convolutional RBMs for Vision. Technical Report 1327. 58, 106

Eslami, S. M. A., Heess, N., Williams, C. K. I., and Winn, J. (2013). The Shape Boltzmann Machine: a Strong Model of Object Shape. *International Journal of Computer Vision (IJCV) (under review)*.

Eslami, S. M. A., Heess, N., and Winn, J. (2012). The Shape Boltzmann Machine: a Strong Model of Object Shape. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Eslami, S. M. A. and Williams, C. K. I. (2011). Factored Shapes and Appearances for Parts-based Object Understanding. In *British Machine Vision Conference (BMVC)*.

Eslami, S. M. A. and Williams, C. K. I. (2012). A Generative Model for Parts-based Object Segmentation. In *Advances in Neural Information Processing Systems 25*.

Everingham, M., Gool, L. V., Williams, C. K. I., Winn, J., and Zisserman, A. (2010). The PASCAL Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision*, 88:303–338. 3, 106

Fei-Fei, L., Fergus, R., and Perona, P. (2004). Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories. In *IEEE Conference on Computer Vision and Pattern Recognition 2004, Workshop on Generative-Model Based Vision*. 44, 62, 79, 113, 118, 120

Felzenszwalb, P. F., Girshick, R. B., McAllester, D., and Ramanan, D. (2009). Object Detection with Discriminatively Trained Part Based Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 99:1–19. 19, 107

Felzenszwalb, P. F. and Huttenlocher, D. P. (2000). Efficient Matching of Pictorial Structures. In *IEEE Conference on Computer Vision and Pattern Recognition 2000*, pages 66–73. 19

Fergus, R., Perona, P., and Zisserman, A. (2003). Object class recognition by unsupervised scale-invariant learning. In *IEEE Conference on Computer Vision and Pattern Recognition 2003*, pages 264–271. 19, 45

Ferrari, V., Jurie, F., and Schmid, C. (2010). From images to shape models for object detection. *International Journal of Computer Vision*, 87(3):284–303. 50

Fischler, M. and Elschlager, R. (1973). The Representation and Matching of Pictorial Structures. *IEEE Transactions on Computers*, 22(1):67–92. vii, 19, 20

Forsyth, D. A. and Ponce, J. (2011). *Computer Vision: A Modern Approach (2nd Edition)*. Prentice Hall Professional Technical Reference. 8

Freund, Y. and Haussler, D. (1994). Unsupervised learning of distributions on binary vectors using two layer networks. Technical Report UCSC-CRL-94-25, University of California, Santa Cruz. 54, 55, 56, 59

Frey, B., Jojic, N., and Kannan, A. (2003). Learning appearance and transparency manifolds of occluded objects in layers. In *IEEE Conference on Computer Vision and Pattern Recognition 2003*, pages 45–52. vii, 3, 17, 19, 26, 36, 49, 107

Friedman, J., Hastie, T., and Tibshirani, R. (2010). Regularization Paths for Generalized Linear Models via Coordinate Descent. *Journal of Statistical Software*, 33(1):1–22. 80

Fulkerson, B., Vedaldi, A., and Soatto, S. (2009). Class Segmentation and Object Localization with Superpixel Neighborhoods. In *International Conference on Computer Vision 2009*, pages 670–677. vii, 14, 15, 26

Gavrila, D. M. (2007). A Bayesian, Exemplar-Based Approach to Hierarchical Shape Matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29:1408–1421. 50, 52

Geman, S. and Geman, D. (1984). Stochastic Relaxation, Gibbs Distributions, and

the Bayesian Restoration of Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741. 13

Graham, D. and Allinson, N. (1998). *Face Recognition: From Theory to Application*, volume 163, chapter Characterizing Virtual Eigensignatures for General Purpose Face Recognition, pages 446–456. 43

He, X., Zemel, R., and Ray, D. (2006). Learning and incorporating top-down cues in image segmentation. In *European Conference on Computer Vision 2006*, volume 1, pages 338–351. 14, 26

Heess, N. (2011). *Learning generative models of mid-level structure in natural images*. PhD thesis, University of Edinburgh. 91

Heess, N., Le Roux, N., and Winn, J. (2011). Weakly Supervised Learning of Foreground-Background Segmentation using Masked RBMs. In *International Conference on Artificial Neural Networks 2011*. 37, 80, 82

Hinton, G. (2002). Training Products of Experts by Minimizing Contrastive Divergence. *Neural Computation*, 14(8):1771–1800. 56

Hoiem, D., Rother, C., and Winn, J. (2007). 3D LayoutCRF for Multi-View Object Class Recognition and Segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition 2007*. 106

Horn, B. (1977). AI: The Tumultuous History of the Search for Artificial Intelligence. *Artificial Intelligence*, 8:201–231. 2

Jojic, N. and Caspi, Y. (2004). Capturing Image Structure with Probabilistic Index Maps. In *IEEE Conference on Computer Vision and Pattern Recognition 2004*, pages 212–219. viii, 23, 24, 52, 92

Jojic, N., Perina, A., Cristani, M., Murino, V., and Frey, B. (2009). Stel component analysis: Modeling spatial correlations in image class structure. In *IEEE Conference on Computer Vision and Pattern Recognition 2009*, pages 2044–2051. 17, 26, 38, 48, 92

Kannan, A., Winn, J., and Rother, C. (2006). Clustering Appearance and Shape by Learning Jigsaws. In *Advances in Neural Information Processing Systems 19*, pages 657–664. 19, 23

Kapoor, A. and Winn, J. (2006). Located Hidden Random Fields: Learning Discrim-

inative Parts for Object Detection. In *European Conference on Computer Vision 2006*, pages 302–315. viii, 23, 24, 26, 85

Kivinen, J. J. and Williams, C. K. I. (2011). Transformation Equivariant Boltzmann Machines. In Honkela, T., Duch, W., Girolami, M. A., and Kaski, S., editors, *International Conference on Artificial Neural Networks 2011*, volume 6791 of *Lecture Notes in Computer Science*, pages 1–9. Springer. 107

Kohli, P., Kumar, M. P., and Torr, P. H. S. (2007). P3 & Beyond: Solving Energies with Higher Order Cliques. In *IEEE Conference on Computer Vision and Pattern Recognition 2007*. 50

Kohli, P., Ladicky, L., and Torr, P. H. S. (2009). Robust Higher Order Potentials for Enforcing Label Consistency. *International Journal of Computer Vision*, 82(3):302–324. 50

Koller, D., Weber, J., and Malik, J. (1994). Robust Multiple Car Tracking with Occlusion Reasoning. In *European Conference on Computer Vision 1994*, pages 185–196. 25

Komodakis, N. and Paragios, N. (2009). Beyond pairwise energies: Efficient optimization for higher-order MRFs. In *IEEE Conference on Computer Vision and Pattern Recognition 2007*, pages 2985–2992. 50

Kumar, M. P., Torr, P. H. S., and Zisserman, A. (2004). Learning layered pictorial structures from video. In *The Indian Conference on Computer Vision, Graphics and Image Processing 2004*, pages 148–153. 20

Kumar, P., Torr, P., and Zisserman, A. (2005). OBJ CUT. In *IEEE Conference on Computer Vision and Pattern Recognition 2005*, pages 18–25. vii, 19, 20, 26, 50

Kumar, S. and Hebert, M. (2003). Discriminative Random Fields: A Discriminative Framework for Contextual Interaction in Classification. In *International Conference on Computer Vision 2003*, pages 1150–1157. 13

Lafferty, J. D., McCallum, A., and Pereira, F. C. N. (2001). Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *International Conference on Machine Learning 2001*, pages 282–289. 13

Larlus, D., Verbeek, J., and Jurie, F. (2009). Category level object segmentation by

combining bag-of-words models with Dirichlet processes and random fields. *International Journal of Computer Vision*, 88:238–253. 20, 26

Le Roux, N. and Bengio, Y. (2008). Representational Power of Restricted Boltzmann Machines and Deep Belief Networks. *Neural Computation*, 20(6):1631–1649. 56

Le Roux, N., Heess, N., Shotton, J., and Winn, J. (2011). Learning a Generative Model of Images by Factoring Appearance and Shape. *Neural Computation*, 23(3):593–650. 25, 37, 49, 80, 87, 105

Lee, H., Grosse, R., Ranganath, R., and Ng, A. Y. (2009). Convolutional Deep Belief Networks for Scalable Unsupervised Learning of Hierarchical Representations. In *International Conference on Machine Learning 2009*, pages 609–616. 58, 75, 107

Leibe, B., Leonardis, A., and Schiele, B. (2004). Combined Object Categorization and Segmentation With An Implicit Shape Model. In *ECCV Workshop on Statistical Learning in Computer Vision*. vii, 21, 22, 26

Lempitsky, V., Kohli, P., Rother, C., and Sharp, T. (2009). Image Segmentation with A Bounding Box Prior. In *International Conference on Computer Vision 2009*, pages 277–284. 14, 26

Levin, A. and Weiss, Y. (2009). Learning to Combine Bottom-Up and Top-Down Segmentation. *International Journal of Computer Vision*, 81:105–118. vii, 21, 22, 26

Li, Y., Tarlow, D., and Zemel, R. (2013). Exploring Compositional High Order Pattern Potentials for Structured Output Learning. In *IEEE Conference on Computer Vision and Pattern Recognition 2013*. 54

Lowe, D. (2004). Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60:91–110. 11, 12, 119

Lücke, J., Turner, R., Sahani, M., and Henniges, M. (2009). Occlusive Components Analysis. In *Advances in Neural Information Processing Systems 22*, pages 1069–1077. 25, 105

Mikolajczyk, K. and Schmid, C. (2004). Scale and affine invariant interest point detectors. *International Journal of Computer Vision*, 60(1):63–86. 119, 120

Morris, R. D., Descombes, X., and Zerubia, J. (1996). The Ising/Potts model is not well

suited to segmentation tasks. In *Proceedings of the IEEE Digital Signal Processing Workshop*. 53

Murray, I., Adams, R. P., and MacKay, D. J. (2010). Elliptical slice sampling. *Journal of Machine Learning Research*, 9:541–548. 35

Murray, I. and Salakhutdinov, R. (2009). Evaluating probabilities under high-dimensional latent variable models. In *Advances in Neural Information Processing Systems 21*. 71

Nabney, I. (2002). *NETLAB: algorithms for pattern recognition*. Springer-Verlag New York, Inc., New York, NY, USA. 117

Neal, R. M. (1992). Connectionist learning of belief networks. *Artificial Intelligence*, 56:71–113. 60

Neal, R. M. (2001). Annealed importance sampling. *Statistics and Computing*, 11(2):125–139. 71

Norouzi, M., Ranjbar, M., and Mori, G. (2009). Stacks of convolutional Restricted Boltzmann Machines for shift-invariant feature learning. In *IEEE Conference on Computer Vision and Pattern Recognition 2009*, pages 2735–2742. 58

Nowozin, S. and Lampert, C. H. (2009). Global connectivity potentials for random field models. In *IEEE Conference on Computer Vision and Pattern Recognition 2009*, pages 818–825. 50, 52

Oren, M., Papageorgiou, C., Sinha, P., Osuna, E., and Poggio, T. (1997). Pedestrian Detection Using Wavelet Templates. In *IEEE Conference on Computer Vision and Pattern Recognition 1997*, pages 193–99. 43

Peterson, M. and Gibson, B. (1993). Shape recognition inputs to figure-ground organization in three-dimensional displays. *Cognitive Psychology*, 25:383–429. 15

Poggio, T., Torre, V., and Koch, C. (1985). Computational vision and regularization theory. *Nature*, 317(26):314–319. 2

Raina, R., Madhavan, A., and Ng, A. Y. (2009). Large-scale deep unsupervised learning using graphics processors. In *International Conference on Machine Learning 2009*, pages 873–880. 58

Ranzato, M., Mnih, V., and Hinton, G. E. (2010). How to Generate Realistic Images

Using Gated MRFs. In *Advances in Neural Information Processing Systems 23*. 58, 106

Ranzato, M., Susskind, J., Mnih, V., and Hinton, G. E. (2011). On deep generative models with applications to recognition. In *IEEE Conference on Computer Vision and Pattern Recognition 2011*, pages 2857–2864. 58

Robbins, H. and Monro, S. (1951). A Stochastic Approximation Method. *The Annals of Mathematical Statistics*, 22(3):400–407. 60

Ross, D. and Zemel, R. (2006). Learning Parts-Based Representations of Data. *Journal of Machine Learning Research*, 7:2369–2397. 23, 37

Roth, S. and Black, M. J. (2005). Fields of Experts: A Framework for Learning Image Priors. In *IEEE Conference on Computer Vision and Pattern Recognition 2005*, pages 860–867. 106

Rother, C., Kohli, P., Feng, W., and Jia, J. (2009). Minimizing sparse higher order energy functions of discrete variables. In *IEEE Conference on Computer Vision and Pattern Recognition 2009*, pages 1382–1389. 50, 54

Rother, C., Kolmogorov, V., and Blake, A. (2004). "GrabCut": interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics (SIGGRAPH) 2004*, 23:309–314. vii, 3, 14, 16, 26, 52, 92

Rowley, H., Baluja, S., and Kanade, T. (1998). Neural Network-Based Face Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1):23–38. 107

Russell, B., Torralba, A., Murphy, K., and Freeman, W. (2008). LabelMe: A Database and Web-Based Tool for Image Annotation. *International Journal of Computer Vision*, 77:157–173. 93, 97

Salakhutdinov, R. and Hinton, G. (2009). Deep Boltzmann Machines. In *International Conference on Artificial Intelligence and Statistics 2009*, volume 5, pages 448–455. 9, 51, 57, 59, 60, 61, 63, 71

Salakhutdinov, R. and Murray, I. (2008). On the Quantitative Analysis of Deep Belief Networks. In *International Conference on Machine Learning 2008*. 71

Schneiderman, H. (2000). *A Statistical Approach to 3D Object Detection Applied to Faces and Cars*. PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA. 107

Shekhovtsov, A., Kohli, P., and Rother, C. (2012). Curvature Prior for MRF-Based Segmentation and Shape Inpainting. In *DAGM/OAGM Symposium*, pages 41–51. 48

Shotton, J., Winn, J., Rother, C., and Criminisi, A. (2006). TextonBoost: Joint Appearance, Shape and Context Modeling for Multi-Class Object Recognition and Segmentation. In *European Conference on Computer Vision 2006*, pages 1–15. 14, 26

Sigal, L., Balan, A., and Black, M. (2010). HumanEva: Synchronized Video and Motion Capture Dataset and Baseline Algorithm for Evaluation of Articulated Human Motion. *International Journal of Computer Vision*, 87(1-2):4–27. 93

Sudderth, E. and Jordan, M. (2008). Shared Segmentation of Natural Scenes Using Dependent Pitman-Yor Processes. In *Advances in Neural Information Processing Systems 22*. 37

Sudderth, E., Torralba, A., Freeman, W., and Willsky, A. (2008). Describing Visual Scenes Using Transformed Objects and Parts. *International Journal of Computer Vision*, 77:291–330. 19

Thomas, A., Ferrari, V., Leibe, B., Tuytelaars, T., and Gool, L. V. (2009). Using Multi-view Recognition and Meta-data Annotation to Guide a Robot's Attention. *International Journal of Robotics Research*, 28(8):976–998. 85, 92, 97, 102

Thomas, A., Ferrari, V., Leibe, B., Tuytelaars, T., Schiele, B., and Gool, L. J. V. (2006). Towards Multi-View Object Class Detection. In *IEEE Conference on Computer Vision and Pattern Recognition 2006*, pages 1589–1596. 106

Tieleman, T. (2008). Training restricted Boltzmann machines using approximations to the likelihood gradient. In *International Conference on Machine Learning 2008*, pages 1064–1071. 56, 60, 61

Tjelmeland, H. and Besag, J. (1998). Markov Random Fields with Higher-Order Interactions. *Scandinavian Journal of Statistics*, 25(3):415–433. 53

Torralba, A., Murphy, K. P., and Freeman, W. T. (2007). Sharing Visual Features for Multiclass and Multiview Object Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29:854–869. 7, 19

Ullman, S. (1996). *High-Level Vision: Object Recognition and Visual Cognition*. MIT Press. 6

Williams, C. K. I. and Titsias, M. (2004). Greedy learning of multiple objects in images using robust statistics and factorial learning. *Neural Computation*, 16(5):1039–1062. 3, 25, 36, 49, 107

Winn, J. and Jojic, N. (2005). LOCUS: Learning object classes with unsupervised segmentation. In *International Conference on Computer Vision 2005*, pages 756–763. vii, 17, 18, 26, 37, 38, 43, 45, 47, 48, 52, 85, 92

Winn, J. and Shotton, J. (2006). The Layout Consistent Random Field for Recognizing and Segmenting Partially Occluded Objects. In *IEEE Conference on Computer Vision and Pattern Recognition 2006*, pages 37–44. 23, 26

Younes, L. (1999). On The Convergence Of Markovian Stochastic Algorithms With Rapidly Decreasing Ergodicity Rates. In *Stochastics and Stochastics Reports*, volume 65, pages 177–228. 60

Younes, L. and Sud, P. (1989). Parametric Inference for imperfectly observed Gibbsian fields. *Probability Theory and Related Fields*, 82:625–645. 60

Zhu, L., Chen, Y., Torralba, A., Yuille, A., and Freeman, W. (2010). Latent Hierarchical Structural Learning for Object Detection. In *IEEE Conference on Computer Vision and Pattern Recognition 2010*. 19