http://www.psi.toronto.edu

Bayesian Painting by Numbers: Flexible Priors for Colour-Invariant Object Recognition

Jeroen C. Chua, Inmar E. Givoni, Ryan P. Adams, Brendan J. Frey

June 9, 2011 **PSI TR 2011–001**

Abstract

Generative models of images should take into account transformations of geometry and reflectance. Then, they can provide explanations of images that are factorized into intrinsic properties that are useful for subsequent tasks, such as object classification. It was previously shown how images and objects within images could be described as compositions of regions called structural elements or 'stels'. In this way, transformations of the reflectance and illumination of object parts could be accounted for using a hidden variable that is used to 'paint' the same stel differently in different images. For example, the stel corresponding to the petals of a flower can be red in one image and yellow in another. Previous stel models have used a fixed number of stels per image and per image class. Here, we introduce a Bayesian stel model, the *colour-invariant admixture* (CIA) model, which can infer different numbers of stels for different object types, as appropriate. Results on Caltech101 images show that this method is capable of automatically selecting a number of stels that reflects the complexity of the object class and that these stels are useful for object recognition.

Bayesian Painting by Numbers: Flexible Priors for Colour-Invariant Object Recognition

Jeroen C. Chua, Inmar E. Givoni, Ryan P. Adams, Brendan J. Frey
University of Toronto

1 Introduction

Vision can be thought of as inference in a learnt model of the relationships between spatial patterns at different levels of abstraction. Marr described three levels of visual patterns: the primal sketch, corresponding to what an artist would draw to represent parts of objects in a scene; the 2.5D sketch, which overlays the primal sketch with textures, colours and shading; and the 3D model, which relates primal and 2.5D sketches derived from different viewpoints and 3D manipulations. Two extreme approaches to developing visual learning algorithms include using highly flexible, unstructured neural networks (Hinton; Hinton and Salakhutdinov, 2006; Serre et al., 2007; Erhan et al., 2010), and using highly structured techniques that hard-wire sensible rules for pattern generation (Grenander; Mumford; Zhu and Mumford, 1998). In the case of neural network approaches, the hope is that Marr's different levels of patterns will emerge after learning in a deep neural network, because they are the most efficient way to model the statistics of images (Hinton; Hinton and Salakhutdinov, 2006). In the case of methods using hard-wired pattern rules, the hope is that a reasonably simple set of rules can be combined with a straightforward inference algorithm to accurately describe the huge variation seen in natural images (Mumford).

We take an approach that combines the best aspects of the neural network and pattern rule approaches, by exploring highly flexible statistical models that incorporate sensible pattern rules. The best known example of this approach is the convolutional neural network (LeCun et al., 1998), which takes an image as input, propagates signals through multiple layers of hidden variables, and then predicts the class of the object in the image. The layers of variables are arranged according to the topology of the input image, and each hidden variable receives input only from nearby variables in the previous layer. This method achieves state-of-the-art performance on several standard classification tasks (LeCun et al., 1998). In our approach, we recognize that in general the number of labels that are available for training is exponentially smaller than the number of possible pattern combinations. Therefore, we use statistical models of the image data itself and train these models in an unsupervised fashion.

Fig. 1 illustrates our approach, which was first described in (Frey and Jojic, 1999; Jojic and Frey, 2001; Frey and Jojic, 2003; Jojic et al., 2009a). Variations due to object location, orientation, scale, reflectance and illumination are factored out and represented in a transformation variable T, and unsupervised learning methods are used to model the normalized, latent image z. From a generative point of view, each image in the dataset is assumed to have been produced by generating a latent image from the model $p(z \,|\, \theta)$, randomly selecting a transformation T from $p(T)$, and then applying the transformation to obtain the observed image according to the rendering model $p(x \,|\, T, z)$. When an object is most naturally described as a composition of articulating, deformable parts, the transformation T should be factorized into a field of transformations where each sub-transformation transforms an object part.

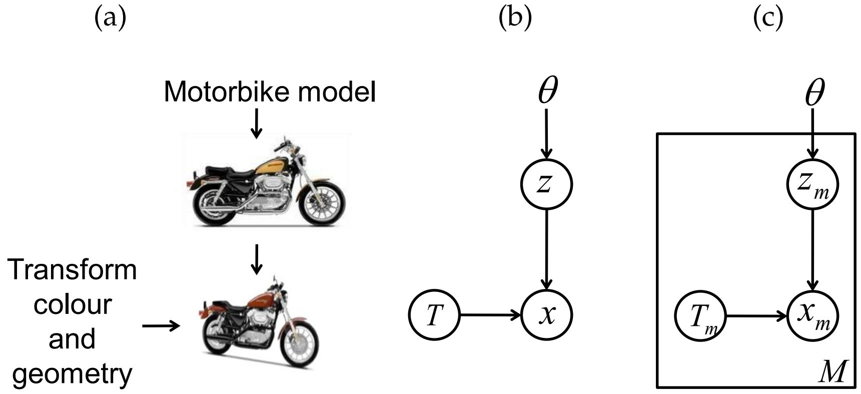


Figure 1: **Factorizing image explanations into intrinsic transformations.** (a) A model is used to generate a description of the appearance of a motorbike, which is then modified by transformations of geometry and reflectance to produce an instance. (b) This process can be formalized by thinking of the observed image x as a random variable. It is assumed to have been generated by applying an image-specific transformation T to a latent image z so as to change various object properties, including reflectance and geometry. The parameter θ describes the distribution over normalized latent images, z. (c) A dataset of M images $x_1, \ldots, x_M$ is generated using corresponding latent images and transformations. Here, plate notation is used, where variables within the box are replicated M times corresponding to the M images.

Here we attend to the vision problem of accounting for variability in reflectance properties and illumination across object instances, so we will assume that all instances of an object have similar geometry. Fig. 2(a) shows two motorbike images from the Caltech101 dataset (Li et al., 2004). The motorbikes and the parts comprising them have similar geometry, but quite different reflectance and illumination properties. For example, a prominent difference between the two images is the colour of the pipework; whereas the first motorbike has black pipework, the second motorbike has light chrome pipework.

A popular standard approach to reducing sensitivity to variations in reflectance and illumination is to pre-whiten images so as to emphasize edges (Olshausen and Field, 1996). Absolute pixel intensities are discarded and instead only information about edges (Canny) or oriented intensity gradients, such as those encoded by SIFT features (Lowe), are used. This approach produced state-of-the-art results on image classification problems in the first decade of this century (Lazebnik et al., 2006). However, it is sensitive to parameters such as the edge detector sensitivity, the patch size used to define SIFT features, thresholds on minimum gradients, the degree of contrast normalization, and so on. Figs. 2(b) and 2(c) illustrate the difficulty in selecting thresholds for a Canny edge detector (Canny). The more stringent threshold used in Fig. 2(b) leads to important edges being lost in regions of low contrast, such as the gas tank in the top image and the pipework in the bottom image. The more liberal threshold used in Fig. 2(c) leads to a large number of spurious edges. The essential problem here is that it is not possible to define beforehand an edge detection threshold that will differentiate all object parts without also erroneously detecting edges due to noise.

A quite different approach to building robustness against variation in reflectance and illumination was proposed in (Jojic et al., 2009b,a). They defined a 'structure element' (stel) as a class-specific region in the image plane that can have different texture, shading and colour in different examples, but whose spatial structure is similar across images. Stels correspond to regions in Marr's primal sketch, which can be rendered differently in different 2.5D sketches.

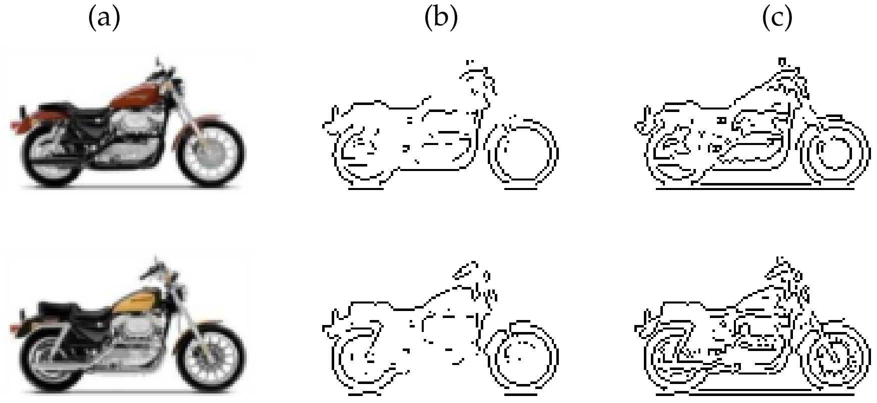


Figure 2: **Attempting to detect object parts using image gradients.** Two Caltech101 images (a) were processed using a Canny edge detector with stringent (b) and liberal (c) thresholds. Object parts are either not delineated or are delineated but accompanied by spurious irrelevant edges.

Stels are identified by indices, so that the latent image z in Fig. 1 is an image of stel indices. The latent image model $p(z\,|\,\theta)$ provides a distribution over index maps. For the current image, T is a colour model that specifies a distribution over colours for each stel index. Given the current index map and colour model, a distribution over colours is specified for every pixel.

Stel models account for appearance in a way that factorizes out instance-specific reflectance and illumination properties. Given a training set of images, the learnt stel model segments images from an object class into different regions (stels) in a colour-invariant way by modelling the co-occurrence of colours within an image and spatial relationships across images within the object class. Pixels that are typically the same colour and can be grouped into similar shapes across images are put into a single stel, which loosely corresponds to an object part. Grouping pixels in this way provides a class-specific bias for parts-based segmentation of training and test images. In the context of object recognition, stel models provide a means to model spatial relationships between oriented gradient features (Jojic et al., 2009b; Perina et al., 2010).

Using the expectation-maximization (EM) algorithm described in (Jojic et al., 2009b), a single-class model with nine stels was learnt from the five Caltech101 translation- and scale-normalized images shown in Fig. 3(a). The image sizes were 75×132 and they were converted to greyscale for analysis. Fig. 3(b) shows the nine stels, where for each stel an image of probabilities that pixels belong to the stel are shown, with white corresponding to a probability of one and black corresponding to a probability of zero. Some stels, such as the stel in the middle row on the left, account for large portions of pixels. Other stels, such as the one in the middle that accounts for the front wheel disk, account for small portions of pixels. The three dominant stels are shown in the left column.

Since stels are defined in a way that is similar to Marr's definition of the primal sketch, an interesting question is whether the learnt stels can be used as a primal sketch. Recall that the problem with the image-derived edge maps shown in Figs. 2(b) and 2(c) is that it is not possible to pick a threshold that yields an edge map that clearly delineates objects and parts, while at the same time not introducing many erroneous edges. Since stels are required to be consistent across images, can they be used to make primal sketches that account for object parts? To answer this question, the three dominant stels were smoothed using a Gaussian filter with $\sigma = 1.5$ and the MATLAB Canny edge detector was applied using default settings. The resulting edge maps are shown in Fig. 4 and correspond to the overall outline of the motorbike,

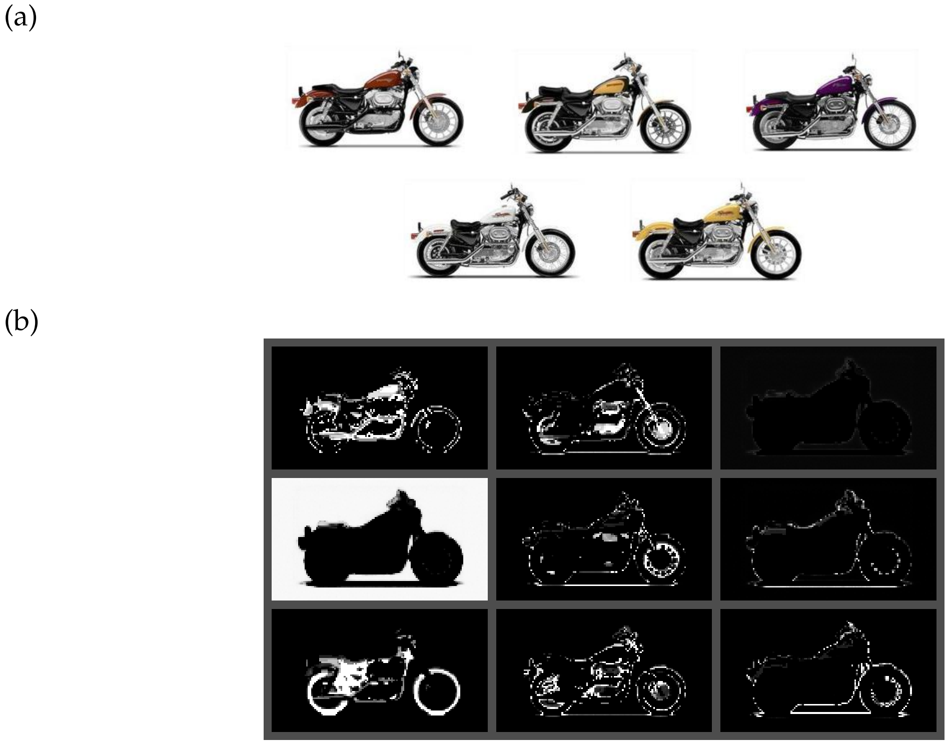


Figure 3: **The structure elements (stels) of motorbike images.** Five images from the Caltech101 database (a) were used to learn nine stels (b). For each stel, an image of probabilities that pixels belong to the stel is shown. The first stel accounts for the pipework, gas tank and rear fender.

the wheels and seat, and the pipework and gas tank.

Since stel segmentations capture interesting spatial relationships within an object class, they can be used in an object recognition framework by using the segmentations to encode the spatial configuration of local image features. It has been previously shown that recognition performance can be improved by incorporation of the spatial relationships between features (Perina et al., 2010), in contrast to models based on bags of visual words derived from, e.g., SIFT features.

A major drawback of current approaches to modelling with stels is that they require the number of stels — the number of object parts for an object class — to be fixed in advance. As it is difficult to determine a class-appropriate number of stels *a priori*, this is an undesirable requirement. Using too few stels may result in segmentations that are too coarse, while using an excessive number of stels may lead to overfitting. Furthermore, differing poses and lighting conditions may call for a different number of stels even within a single object class. This important free parameter has typically been set by hand or by using computationally-expensive cross-validation.

Here, we propose a Bayesian stel model that uses a prior distribution over the assignment of pixels to stels to regularize the complexity of the stel segmentation. After learning, the posterior distribution captures information about the appropriate distribution over stels for a given set of data. We develop a framework for stels that models images as an admixture, complementing other approaches, such as latent Dirichlet allocation (Blei et al., 2003; Sudderth et al., 2008; Sivic et al., 2005; Cao and Li, 2007).

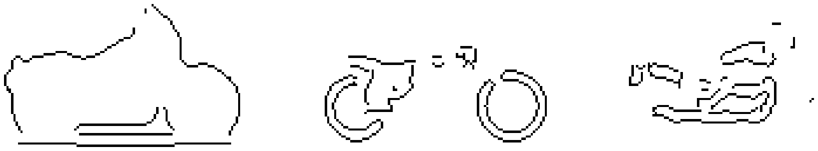


Figure 4: **Primal sketches derived from stels.** These three edge maps were obtained from smoothed versions of the three dominant stels shown in the left column of Fig. 3(b). They correspond to 'primal sketches' of the motorbike outline, the wheels and seat, and the pipework and gas tank.

2 The Colour-Invariant Admixture Model

One powerful approach to modelling data is to use an admixture, which captures the idea that a given datum (e.g., an image) may be a combination of several latent components. This idea has found wide use in the modelling of natural language documents, where latent Dirichlet allocation (LDA) (Blei et al., 2003) provides a particularly convenient and elegant generative probabilistic model for exchangeable text data. When considering the problem of vision from a modelling point of view, Marr's notion of a primal sketch maps well onto the admixture concept. Considering again the stel-derived edge maps in Fig. 4, we can imagine that these sketches are blended together to produce the observed image. Note that this is in contrast to a simple mixture model, where images would result from precisely one of these three sketches.

In this section we develop a generative Bayesian variant of the stel model, which we call the *colour-invariant admixture* (CIA) model. This model extends the standard approach to stel modelling to enable representation of the full posterior distribution over the stels. By combining the powerful ability to learn spatial relationships using stels, with the flexible invariance properties of a fully-probabilistic latent colour model, CIA is able to learn image-specific properties of colour that enable richer feature extraction for supervised learning tasks. Inference is straightforward, using Markov chain Monte Carlo.

2.1 The Standard Stel Model

We first formally describe the standard stel model, as outlined in (Jojic et al., 2009b). We assume that there are M images, each with N pixels. We denote the nth pixel in the mth image by $x_{n,m}$, taking values in a colour space $\mathcal{C}$, which we will take without loss of generality to be $\mathbb{R}^D$. We denote the collection of all pixel values across all images by $\mathbf{X} = \{\{x_{n,m}\}_{n=1}^N\}_{m=1}^M$.

The standard stel approach associates with each pixel measurement $x_{n,m}$ a discrete variable $s_{n,m} \in \{1, 2, \dots, S\}$, which indicates the stel to which that pixel belongs. We denote this set of indices as $\Xi = \{\{s_{n,m}\}_{n=1}^N\}_{m=1}^M$. A stel can be loosely thought of as either a background model, or an object part. For instance, for the motorcycle class, one stel may represent the wheels, another stel may represent the pipework, and another stel may represent the background.

The main assumption of the stel model is that pixels belonging to the same stel have high probability under a tight distribution defined on $\mathcal{C}$. That is, the stel identity of a pixel is highly informative about colour. The key insight is to allow these distributions to vary across images, i.e., in image m, stel s has unique parameters $\phi_{s,m}$. If the observation model $p(x \mid \phi)$ is a Gaussian distribution on $\mathcal{C}$, for example, then $\phi_{s,m}$ would be mean and covariance parameters that are specific to the combination of s and m. We denote the aggregate set of colour-distribution parameters for the training images as $\Phi = \{\{\phi_{s,m}\}_{s=1}^S\}_{m=1}^M$.

The statistical sharing across images occurs through the per-pixel multinomial distribution

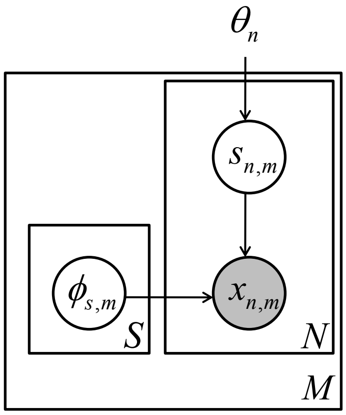


Figure 5: Graphical model for the standard stel model

over the stel assignments, parameterized by $\boldsymbol{\theta}_n$, where $\theta_{s,n} \geq 0$ and $\sum_s \theta_{s,n} = 1$. We denote the aggregate set of index distributions as $\Theta = \{\boldsymbol{\theta}_n\}_{n=1}^N$. The standard stel graphical model is shown in Fig. 5.

Note that since each image can have unique colour distributions, the same object part can have different colours in different images. Therefore, the inferred assignments of stels will be invariant to colour in the sense that what matters is not the *specific* colour, but colour *co-occurrence*. It is not necessary for object parts to be of the same colour in all training images; all that is required is that object parts regularly co-occur in colour across images. Note, however, that the training set must be normalized in position, orientation, and scale, since inference of stel assignments is on a per-pixel basis, and the probability distributions $\boldsymbol{\theta}_n$ are shared across all images. It is possible to add in deformation variables to deal with, e.g., translation and rotation. However, even without such deformation variables, the stel model is capable of handling small amounts of deformation via soft stel assignments (Perina et al., 2010).

Learning of the parameters Θ and Φ can be performed via maximum likelihood using the expectation maximization (EM) algorithm as in (Jojic et al., 2009b). As the pixel-wise stel distributions $\boldsymbol{\theta}_n$ are shared across the entire training set and are invariant to colour, the stel model can provide a robust prior distribution over segmentation for a given object class. Given a test image, the posterior stel segmentation can be efficiently inferred, and this segmentation can be used for other tasks such as object recognition.

2.2 The Stel Model as a Generative Admixture

In practice, the performance of the stel model is very sensitive to the regularization on Θ. Using too strong of a regularization results in coarse, uninformative image segmentations, but allowing too much flexibility results in overfitting. To handle this difficulty, we propose a Bayesian approach that is capable of maintaining a full posterior distribution over Θ. This helps to relieve overfitting, but still results in a flexible model. At test time, the uncertainty in the stel parameters can be taken into account when performing segmentation and object recognition. Additionally, this approach enables CIA to be used as a module in larger hierarchical models with little modification.

As before, $s_{n,m} \in \{1, 2, \ldots, S\}$ is the stel assignment of pixel n in image m, and $\boldsymbol{\theta}_n$ specifies the multinomial distribution over stel indices for pixel n. We will assume that the pixel observation model is a Gaussian distribution with unknown mean and covariance, i.e., $\phi_{s,m} = \{\mu_{s,m},\ \Lambda_{s,m}\}$. We place a Dirichlet prior on the $\boldsymbol{\theta}_n$ with base measure γ. We place normal-inverse-Wishart (NIW) priors on the $\phi_{s,m}$, with parameters $\alpha = \{\mu_0, \kappa_0, \nu_0, \Lambda_0\}$. This leads to the following

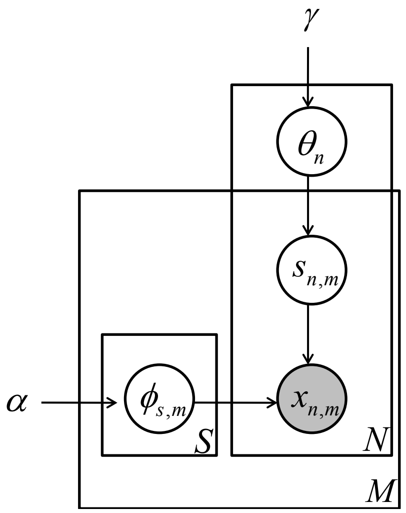


Figure 6: Graphical model for CIA, where s indexes stels, n indexes pixels and m indexes images.

generative model:

$$\boldsymbol{\theta}_n \,|\, \gamma \sim \text{Dirichlet}(\gamma) \tag{1}$$
$$s_{n,m} \,|\, \boldsymbol{\theta}_n \sim \text{Multinomial}(\boldsymbol{\theta}_n, 1) \tag{2}$$
$$\phi_{s,m} \,|\, \alpha \sim \text{NIW}(\mu_0, \kappa_0, \nu_0, \Lambda_0) \tag{3}$$
$$x_{n,m} \,|\, s_{n,m}, \{\phi_{s,m}\}_{s=1}^{S} \sim \text{Norm}(\mu_{s_{n,m},m},\, \Lambda_{s_{n,m},m}). \tag{4}$$

The graphical model for this generative procedure is provided in Fig. 6. In this paper we will use parameterization of the normal-inverse-Wishart given by (Murphy, 2007):

$$\begin{aligned}\text{NIW}(\mu, \Sigma \,|\, \mu_0, \kappa_0, \nu_0, \Lambda_0) = \frac{|\Lambda_0|^{\nu_0/2}}{2^{D\nu_0/2}\Gamma_D(\nu_0/2)(2\pi/\kappa_0)^{D/2}} |\Sigma|^{-(\nu_0+D)/2-1} \\ \times \exp\left\{-\frac{1}{2}\text{tr}\left(\Lambda_0\Sigma^{-1}\right) - \frac{\kappa_0}{2}(\mu-\mu_0)^{\mathsf{T}}\Sigma^{-1}(\mu-\mu_0)\right\},\end{aligned}$$

where $\Gamma_D(\cdot)$ is the generalized gamma function given by

$$\Gamma_D(z) = \pi^{D(D-1)/4} \prod_{d=1}^{D} \Gamma\left(\frac{1}{2}(2z+1-D)\right).$$

In this generative model, we are specifying a full joint distribution over the training images that possesses two different kinds of sharing across the data. In the first case, all of the images (which are assumed to be from a single object class), share the parameter $\boldsymbol{\theta}_n$ that provides the distribution over stels at the pixel level. However, each individual image can use different actual stel assignments to account for variations in object boundaries, deformations, etc. Since the pixels within an image are conditionally independent given Θ and Φ, multiple stels can be represented in a single image. This is the heart of the admixture idea. The second type of sharing is within a single image: each image has its own unique set of distributions associated with its colours. This enables robustness to variation in reflectance and illumination, but supports the intuitive inductive bias that all the pixels for a single stel within an image should have similar properties.

2.3 Inference via Gibbs Sampling

Having specified a stel-based generative model, we can now examine the task of learning, which in this case corresponds to finding the marginal posterior over the stel distributions, given the data and hyperparameters:

$$p(\Theta \,|\, \mathbf{X}, \gamma, \alpha) = \int \mathrm{d}\Phi \int \mathrm{d}\Xi \, p(\Theta, \Phi, \Xi \,|\, \mathbf{X}, \gamma, \alpha). \tag{5}$$

The integrals required to compute this marginal distribution are intractable, however. We therefore take an approximate inference approach based on numerical integration via Markov chain Monte Carlo (MCMC). We begin by taking the generative model specified in Eqs.(1)-(4) and constructing a joint distribution over the data and the unknowns, given the hyperparameters γ and α:

$$p(\mathbf{X}, \Xi, \Theta, \Phi \,|\, \gamma, \alpha) \propto p(\Theta, \Phi, \Xi \,|\, \mathbf{X}, \gamma, \alpha).$$

This distribution is proportional to the posterior distribution over all unknowns that appears inside the integral in Eq. (5). Although we are primarily interested in Θ, by examining the graphical model in Fig. 6, we observe that the posterior distribution over Θ can be easily computed given Ξ and γ. Therefore, it is sufficient for our purposes to generate samples from the posterior distribution over Ξ, marginalizing over both Θ and Φ:

$$p(\Xi \,|\, \mathbf{X}, \gamma, \alpha) \propto p(\mathbf{X}, \Xi \,|\, \gamma, \alpha) = \int \mathrm{d}\Phi \int \mathrm{d}\Theta p(\mathbf{X}, \Xi, \Phi, \Theta \,|\, \gamma, \alpha). \tag{6}$$

We further observe that the distribution in Eq. (6) factorizes as

$$\begin{aligned} &= \int \mathrm{d}\Phi \int \mathrm{d}\Theta \, p(\mathbf{X} \,|\, \Xi, \Phi) \, p(\Xi \,|\, \Theta) \, p(\Theta \,|\, \gamma) \, p(\Phi \,|\, \alpha) \\ &= \left[\int \mathrm{d}\Phi \, p(\mathbf{X} \,|\, \Xi, \Phi) \, p(\Phi \,|\, \alpha) \right] \left[\int \mathrm{d}\Theta \, p(\Xi \,|\, \Theta) \, p(\Theta \,|\, \gamma) \right]. \end{aligned} \tag{7}$$

When constructing the generative stel model, we could have used various priors for $p(\Phi \,|\, \alpha)$ and $p(\Theta \,|\, \gamma)$. However, our specific choice of NIW and Dirichlet distributions, respectively, leads to analytic solutions for the two integral factors in Eq. 7.

In the first case, we have

$$\int \mathrm{d}\Phi \, p(\mathbf{X} \,|\, \Xi, \Phi) \, p(\Phi \,|\, \alpha) = p(\mathbf{X} \,|\, \Xi, \alpha), \tag{8}$$

which we recognize as the marginal likelihood of the data (the denominator of Bayes' theorem), as partitioned by the stel assignments Ξ. Our objective will be to perform a Gibbs sweep over all assignments. To do this, we can factor Eq. (8) into

$$p(\mathbf{X} \,|\, \Xi, \alpha) = p(x_{n^\star, m^\star} \,|\, \mathbf{X}/x_{n^\star, m^\star}, \Xi, \alpha) \, p(\mathbf{X}/x_{n^\star, m^\star} \,|\, \Xi, \alpha).$$

The first factor is the posterior predictive distribution given all data except for pixel $n^\star$ in image $m^\star$. The second term is a constant which does not depend on the pixel we are currently updating. To compute the predictive distribution for the triplet of stel s, pixel n and image m, we begin by finding the sufficient statistics of the data *excluding* pixel n in image m. Let $\delta(s, s')$ be the Kronecker delta function, which takes value one if s and s' are equal and zero otherwise.

The sufficient statistics are then:

$$N_{s,n,m} = \sum_{n' \neq n}^{N} \delta(s, s_{n',m}) \tag{9}$$

$$\bar{x}_{s,n,m} = \frac{1}{N_{s,n,m}} \sum_{n' \neq n}^{N} \delta(s, s_{n',m})\, x_{n',m} \tag{10}$$

$$\bar{\boldsymbol{X}}_{s,n,m} = \frac{1}{N_{s,n,m}} \sum_{n' \neq n}^{N} \delta(s, s_{n',m}) \left[(x_{n',m} - \bar{x}_{s,n,m})(x_{n',m} - \bar{x}_{s,n,m})^{\mathsf{T}} \right]. \tag{11}$$

Following the notation of (Murphy, 2007), these statistics can be used to find the parameters of the normal-inverse-Wishart posterior on the colour distribution:

$$\begin{aligned}
\kappa_{s,n,m} &= \kappa_0 + N_{s,n,m} \\
\mu_{s,n,m} &= \frac{\kappa_0}{\kappa_0 + N_{s,n,m}} \mu_0 + \frac{N_{s,n,m}}{\kappa_0 + N_{s,n,m}} \bar{x}_{s,n,m} \\
\nu_{s,n,m} &= \nu_0 + N_{s,n,m} \\
\Lambda_{s,n,m} &= \Lambda_0 + \bar{\boldsymbol{X}}_{s,n,m} + \frac{\kappa_0\, N_{s,n,m}}{\kappa_0 + N_{s,n,m}} (\bar{x}_{s,n,m} - \mu_0)(\bar{x}_{s,n,m} - \mu_0)^{\mathsf{T}}.
\end{aligned}$$

These parameters also provide a closed form for the posterior predictive distribution, which is a multivariate Student t-distribution:

$$p(x_{n^\star,m^\star} \,|\, \mathbf{X}/x_{n^\star,m^\star}, \Xi, \alpha) = t_{\nu_{s,n,m} - D + 1} \left(\mu_{s,n,m}, \frac{\Lambda_{s,n,m}(\kappa_{s,n,m} + 1)}{\kappa_{s,n,m}(\nu_{s,n,m} - D + 1)} \right),$$

where D is the dimensionality of the colour space, e.g., $D\!=\!3$ for RGB and $D\!=\!1$ for greyscale. The probability density function of the Student t-distribution is given by

$$t_\nu(x \,|\, \mu, \Sigma) = \frac{\Gamma(\nu/2 + D/2)}{\Gamma(\nu/2)} \frac{|\Sigma|^{-1/2}}{(\pi\nu)^{D/2}} \left(1 + \frac{1}{\nu}(x - \mu)^{\mathsf{T}} \Sigma^{-1} (x - \mu) \right)^{-(\nu+D)/2}. \tag{12}$$

In the second factor of Eq.(7), we are also computing a marginal likelihood:

$$\begin{aligned}
p(\Xi \,|\, \gamma) &= \int \mathrm{d}\Theta\, p(\Xi \,|\, \Theta)\, p(\Theta \,|\, \gamma) \\
&= \int \mathrm{d}\Theta \prod_{n=1}^{N} \frac{\Gamma\left(\sum_{s=1}^{S} \gamma_s\right)}{\prod_{s=1}^{S} \Gamma(\gamma_s)} \prod_{s=1}^{S} \theta_{s,n}^{\gamma_s - 1} \prod_{m=1}^{M} \theta_{s,n}^{\delta(s, s_{n,m})}.
\end{aligned} \tag{13}$$

Introducing the sufficient statistic

$$\eta_{s,n} = \sum_{m=1}^{M} \delta(s, s_{n,m}),$$

we can rewrite Eq. (13) as a Dirichlet-multinomial (or Pólya) distribution:

$$\begin{aligned} p(\Xi \,|\, \gamma) &= \int \mathrm{d}\Theta \prod_{n=1}^{N} \frac{\Gamma\left(\sum_{s=1}^{S} \gamma_s\right)}{\prod_{s=1}^{S} \Gamma(\gamma_s)} \prod_{s=1}^{S} \theta_{s,n}^{\gamma_s + \eta_{s,n} - 1} \\ &= \prod_{n=1}^{N} \int \mathrm{d}\boldsymbol{\theta}_n \frac{\Gamma\left(\sum_{s=1}^{S} \gamma_s\right)}{\prod_{s=1}^{S} \Gamma(\gamma_s)} \prod_{s=1}^{S} \theta_{s,n}^{\gamma_s + \eta_{s,n} - 1} \\ &= \prod_{n=1}^{N} \frac{\Gamma\left(\sum_{s=1}^{S} \gamma_s\right)}{\Gamma\left(\sum_{s=1}^{S} \gamma_s + \eta_{s,n}\right)} \prod_{s=1}^{S} \frac{\Gamma(\gamma_s + \eta_{s,n})}{\Gamma(\gamma_s)}. \end{aligned}$$

In order to Gibbs sample, we require the conditional distribution of a single assignment $s_{n,m}$ given all the rest of Ξ. As in the NIW case, we can factorize the marginal likelihood and compute the predictive sufficient statistics for each stel-image-pixel triplet:

$$\eta_{s,n,m} = \sum_{m' \neq m}^{M} \delta(s, s_{n,m'}) = \eta_{s,n} - \delta(s, s_{n,m}).$$

The statistic $\eta_{s,n,m}$ is the number of images for which pixel n has been assigned to stel s, excluding image m. This leads to the posterior predictive for $s_{n,m}$ given all other assignments:

$$p(s_{n,m} = s \,|\, \Xi / s_{n,m}, \gamma) = \frac{\eta_{s,n,m} + \gamma_s}{\sum_{s'=1}^{S} \eta_{s',n,m} + \gamma_{s'}}. \tag{14}$$

The overall Gibbs sampling update is then the product of the "prior" of the assignment induced by the Dirichlet-multinomial predictive distribution in Eq. (14) and the "likelihood" of the pixel intensity that results from the Student t-distribution:

$$\begin{aligned} &p(s_{n,m} = s \,|\, \Xi / s_{n,m}, \mathbf{X}, \gamma, \alpha) \\ &\qquad \propto \frac{\eta_{s,n,m} + \gamma_s}{\sum_{s'=1}^{S} \eta_{s',n,m} + \gamma_{s'}} \, t_{\nu_{s,n,m} - D + 1}\left(\mu_{s,n,m}, \frac{\Lambda_{s,n,m}(\kappa_{s,n,m} + 1)}{\kappa_{s,n,m}(\nu_{s,n,m} - D + 1)}\right). \end{aligned} \tag{15}$$

Finally, given a sample of Ξ from this Markov chain, we can compute the conditional distribution over the stels Θ. These have a Dirichlet posterior distribution:

$$\begin{aligned} p(\Theta \,|\, \Xi, \gamma) &= \prod_{n=1}^{N} p(\boldsymbol{\theta}_n \,|\, \{s_{n,m}\}_{m=1}^{M}, \gamma) \\ &= \prod_{n=1}^{N} \frac{\Gamma\left(\sum_{s=1}^{S} \eta_{s,n} + \gamma_s\right)}{\prod_{s=1}^{S} \Gamma(\eta_{s,n} + \gamma_s)} \prod_{s=1}^{S} \theta_{s,n}^{\eta_{s,n} + \gamma_s - 1}. \end{aligned} \tag{16}$$

2.4 Estimating The Posterior Distribution over Stels

CIA is an unsupervised learning model of image statistics. For practical discriminative tasks, however, we wish to use CIA to provide informative features. We do this by constructing an estimate of the stel assignment probabilities Θ for each of the object classes we wish to identify.

The richest representation of the posterior distribution is achieved by constructing a mixture of Dirichlet distributions, where each component in the mixture is parameterized as in Eq. (16) and weighted equally:

$$p(\Theta \,|\, \gamma, \alpha) \approx \frac{1}{J} \sum_{j=1}^{J} \prod_{n=1}^{N} \frac{\Gamma\left(\sum_{s=1}^{S} \eta_{s,n}^{(j)} + \gamma_s\right)}{\prod_{s=1}^{S} \Gamma(\eta_{s,n}^{(j)} + \gamma_s)} \prod_{s=1}^{S} \theta_{s,n}^{\eta_{s,n}^{(j)} + \gamma_s - 1},$$

where $\eta_{s,n}^{(j)}$ denotes the jth sample of the assignments in the Markov chain from the previous section.

From the point of view of feature-extraction, however, it may be more practical to simply use a point estimate of Θ, denoted $\widehat{\Theta}$. One straightforward way to form such a point estimate is to average the predictive distributions arising from Eq. (16) as in

$$\hat{\theta}_{s,n} = \sum_{j=1}^{J} \frac{\eta_{s,n}^{(j)} + \gamma_s}{\sum_{s'=1}^{S} \eta_{s',n}^{(j)} + \gamma_{s'}}. \tag{17}$$

This uses the same $\eta_{s,n}^{(j)}$ samples as above. This point estimator is used for the experiments in this paper, although with the additional aspect that γ_s is set to zero for prediction. That is, the Dirichlet prior is used for training, but the predictions are not smoothed. This helps identify which stels are actually represented in the data.

An astute reader will notice, however, that stel indices are non-identifiable, but Eq. (17) implicitly assumes that stel indices are in fact identifiable across samples. However, we note that in practice, after a sufficient burn-in period, stel indices appear to not change in between samples due to the extremely slow mixing of the Gibbs sampler, and so practically, stel indices can be treated as identifiable across samples. A theoretically valid approach would be that of analyzing statistics of the co-occurrence of stel membership assignments.

As we expect, some stels have a very small posterior for a given pixel. In fact, some stels have a very small posterior across all pixels in all images. These stels can be thought of the unused stels and their existence supports the notion that different classes need a different number of stels.

Given the per-pixel posterior distribution, we retain only the stels that are “in use” for the class. We define a stel to be in use if its total posterior distribution over pixels is greater than 0.02. That is, a stel s is in use if

$$\sum_{n=1}^{N} \hat{\theta}_{s,n} > 0.02. \tag{18}$$

If there are C classes of interest, indexed by c, then the posterior for each pixel n in class c is represented as an $S^{(c)}$-dimensional vector, where $S^{(c)}$ is the number of stels that are in use for class c.

Overall, the parameters we extract from the inference procedure across all classes are the collection of $\widehat{\Theta}^{(c)}$ distributions for every pixel in every class, where the dimension of $\widehat{\Theta}^{(c)}$ depends on the number of in use stels for class c. This set of parameters is the output of the CIA model that we will use in order to extract useful image representations for object recognition.

3 Using Stels for Supervised Learning Tasks

Stel models are density models of images and can be used for a wide variety of vision tasks. For image classification, stel models can be used to define class-conditional densities which are combined using Bayes’ rule to classify test images, or stel models can be used to construct feature vectors that are fed into a discriminative learning algorithm. In Sect. 4, we report results on image classification using the latter approach. For this purpose, we construct feature vectors (image descriptors) in a way that is similar to the approach described in (Perina et al., 2010).

Stel models are used to define class-specific segmentations of images and those segmentations are used to construct feature vectors consisting of a histogram of SIFT codewords for each stel. In particular, we extract for any image (training or testing) a descriptor based on the $\widehat{\Theta}^{(c)}$ distributions described above. The descriptor can be easily used to calculate image similarities in a variety of ways. For example, a kernel operator that is based on the histogram intersection kernel (Grauman and Darrell, 2005) can be used to measure image similarity and a maximum-margin method such as the SVM can be used for classification.

We define a discrete set of K visual feature codewords $k \in \{1, 2, \ldots, K\}$. For instance, to learn the codebook we can use the standard approach of extracting dense SIFT features and clustering them. We can then pre-process each image by computing per-pixel visual features $\{f_n \in \{1, 2, \ldots, K\}\}_{n=1}^{N}$.

Now, given the feature index associated with each pixel n in the image, we construct a per-class, per-stel count histogram of visual features:

$$h_s^{(c)}(k) = \sum_{n=1}^{N} \hat{\theta}_{s,n}^{(c)} \delta(f_n, k). \tag{19}$$

We can define a concatenated histogram of features as

$$h^{(c)} = [h_1^{(c)}(1), \ldots, h_1^{(c)}(K), h_1^{(c)}(1), \ldots, h_2^{(c)}(K), \ldots, h_{S^{(c)}}^{(c)}(1), \ldots, h_{S^{(c)}}^{(c)}(K)],$$

where $S^{(c)}$ is the total number of stels used for object class c associated with the image. Thus, this representation gives a vector $h^{(c)} \in \mathbb{R}^{S^{(c)} \times K}$, and we obtain a set of C such vectors per image.

Using the histogram of features $h^{(c)}$, we define the class-specific similarity between two images, A and B, by the histogram intersection kernel (Grauman and Darrell, 2005):

$$\text{Ker}(A, B) = \min_{k} [h_A^{(c)}(k), h_B^{(c)}(k)]. \tag{20}$$

Note that using this stel kernel, one can better encode spatial relations. Rather than collecting histograms over arbitrarily defined quadrants as in (Lazebnik et al., 2006), we collect histograms over stel segmentations, which provide useful spatial clues pertaining to the identity of the object.

An orthogonal representation to the CIA representation above is that of the spatial pyramid histogram of features (Lazebnik et al., 2006), which is created by constructing a two-level spatial pyramid histogram of visual features over the four image quadrants, as well as the entire image. Note that the features can be created by a different codebook from the one used to create the $h^{(c)}$ collection.

We also define a combined descriptor, that is created by appending the aforementioned spatial pyramid descriptor to $h^{(c)}$. Thus, we obtain for every image a collection of C feature vectors, $h^{(c)} \in \mathbb{R}^{S^{(c)} \times K + 5K'}$, where the $5K'$ term comes from the spatial pyramid descriptor. In our experiments, we do use the same codebook, and $K = K'$. This representation can also be used in conjunction with the intersection kernel above to give a per-class kernel based similarity measure.

4 Experimental Evaluation

To evaluate the usefulness of our approach, we conducted several experiments, which we report in this section. We performed qualitative assessment by examining whether our new method, CIA, results in stel-based representations of images with varying levels of complexity as determined by the object class. We also performed quantitative analyses to determine whether CIA results in features that improve performance on object recognition tasks. Finally, we also are interested in how the behavior of CIA changes when the free parameters of the prior are adjusted. We investigated these properties using a subset of the Caltech101 image dataset.

4.1 Experimental Setup

4.1.1 Data

We selected a subset of 28 classes from the popular Caltech101 dataset, by identifying the 28 classes that had the largest number of examples per class. To balance class sizes, we chose 30 images per class randomly. The class labels are aeroplane, motorbike, background_Google, faces_easy, watch, leopard, bonsai, car_side, ketch, chandelier, hawksbill, grand_piano, brain, butterfly, helicopter, menorah, kangaroo, starfish, trilobite, buddha, ewer, sunflower, scorpion, revolver, laptop, ibis, llama and minaret. Multiple training-testing trials were used to obtain confidence intervals. In each trial, the 30 images in each class were randomly split into 15 training images and 15 test images. Experimental results are reported based on averaging ten such trials.

4.1.2 Image preprocessing

The images were resized without cropping to be 50×50, and were converted to greyscale, with the intensities scaled to the interval $[0, 1]$. This corresponds to $D = 1$ in Eq. (12). Note that images were *not* whitened (as is common in other vision approaches), since the CIA approach explicitly addresses invariance to colour.

4.1.3 Model configuration

Unless otherwise specified, the maximum number of stels S was set to 12. The experiments used hyperparameters of $\gamma = 0.8$, $\Lambda_0 = 0.001$, $\mu_0 = 0.5$, $\kappa_0 = 0.05$, and $\nu_0 = 2.5$. The Gibbs sampler was used to generate 1200 samples, after burning in for 1800 iterations.

4.1.4 Visual feature (SIFT) codebook

Following the approach of (Lazebnik et al., 2006), we extracted $100,000$ random SIFT features from the training set using code obtained from (Russell et al., 2008), and learnt a codebook of $K = 300$ visual codewords using K-means.

4.2 Learning a Flexible Number of Stels

Our first concern is whether the additional flexibility of the CIA model actually results in richer representations for different object classes. We gauged this by examining how many stels tend to be represented for each object class, when trained using the same hyperparameters. For each class, we trained a stel model separately and after training, the number of significant stels for each class, called 'stel usage', was determined using the threshold described in Eq. 18. For every class, this procedure was repeated ten times. We then examined the number of used stels per class across the different classes, and we report results averaged across trials.

Fig. 7 shows the mean and standard deviation of stel usage for each class, where the classes are sorted according to mean stel usage. Note that different classes use a different number of stels and this number varies widely across classes. The models that use the smallest number of stels on average correspond to the ibis, leopard and starfish classes, whereas models that use the largest number of stels on average correspond to the minaret, face and laptop classes.

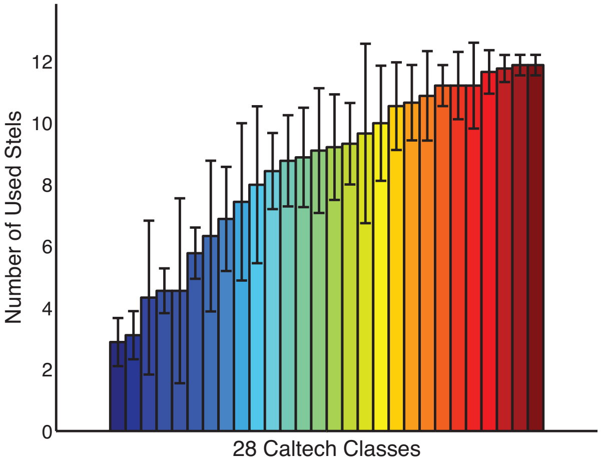
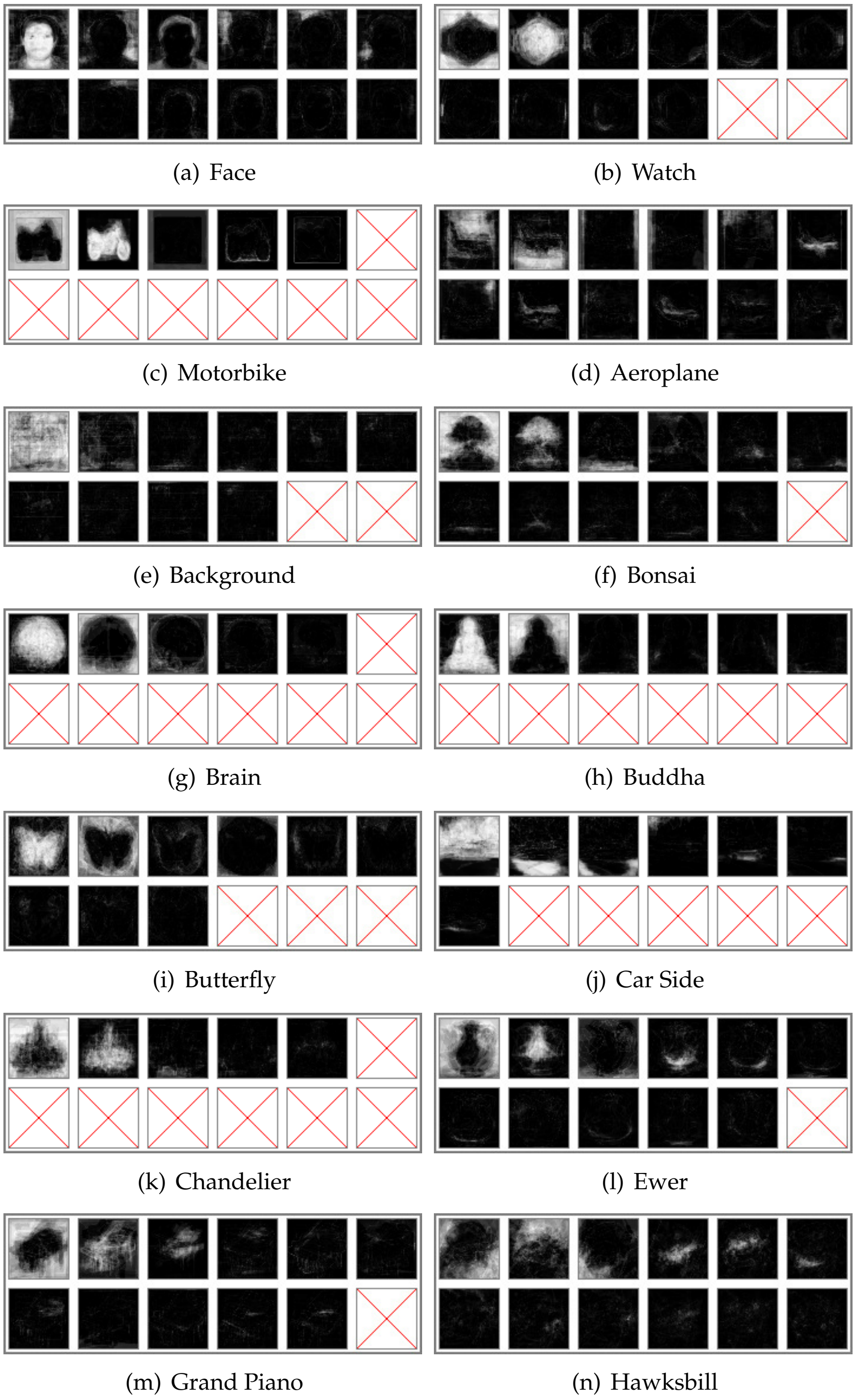


Figure 7: A plot of the number of stels used by different Caltech classes. Error bars show the standard deviation and the classes are sorted by the mean number of stels used.



(a) Face

(b) Watch

(c) Motorbike

(d) Aeroplane

(e) Background

(f) Bonsai

(g) Brain

(h) Buddha

(i) Butterfly

(j) Car Side

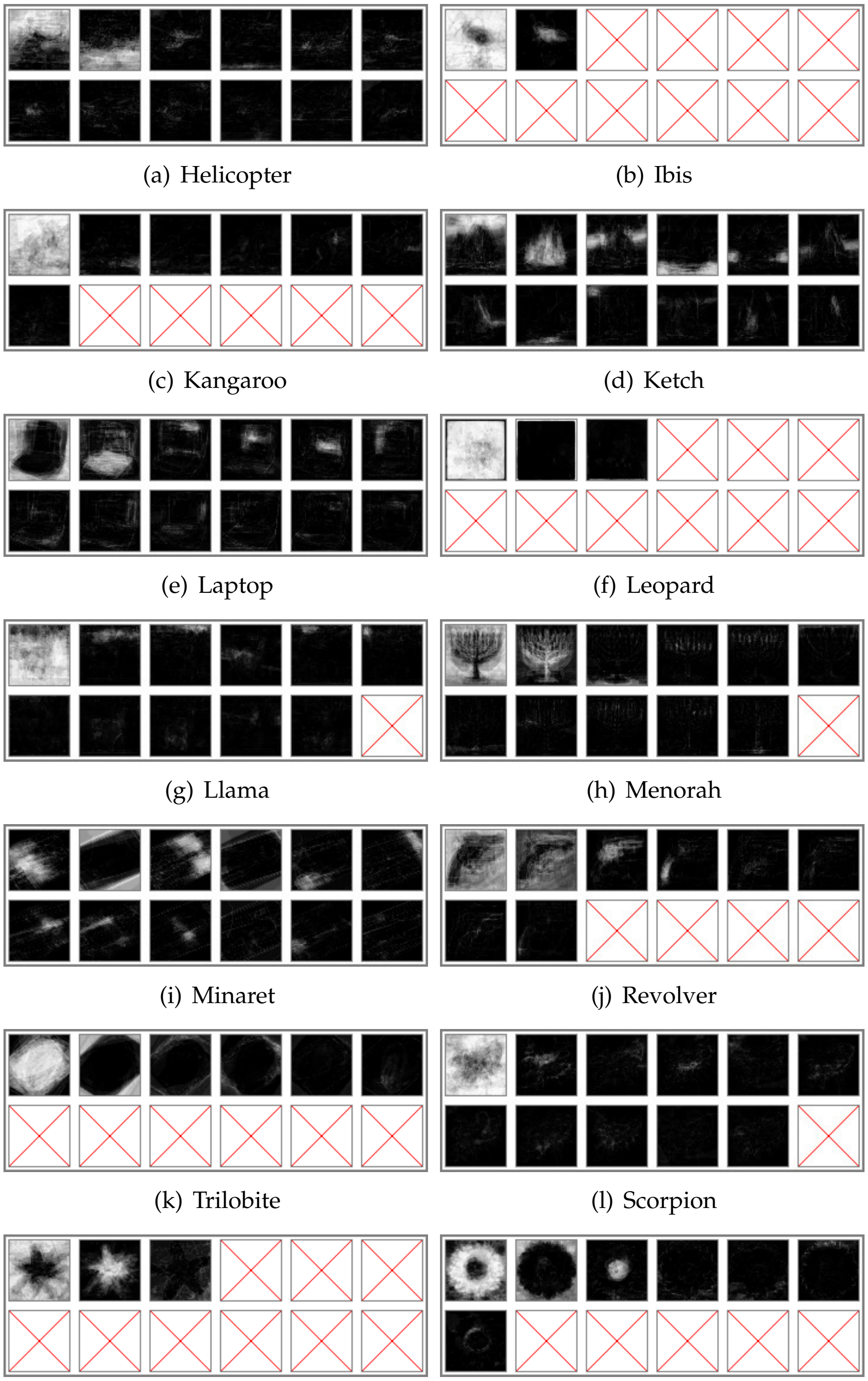
(k) Chandelier

(l) Ewer

(m) Grand Piano

(n) Hawksbill

Figure 8: For each class, the stels learnt in a randomly selected trial are shown in order of decreasing probability mass. Some classes, such as faces, are modelled using a larger number of stels, while other classes, such as chandeliers, are modelled using a smaller number of stels.



(a) Helicopter

(b) Ibis

(c) Kangaroo

(d) Ketch

(e) Laptop

(f) Leopard

(g) Llama

(h) Menorah

(i) Minaret

(j) Revolver

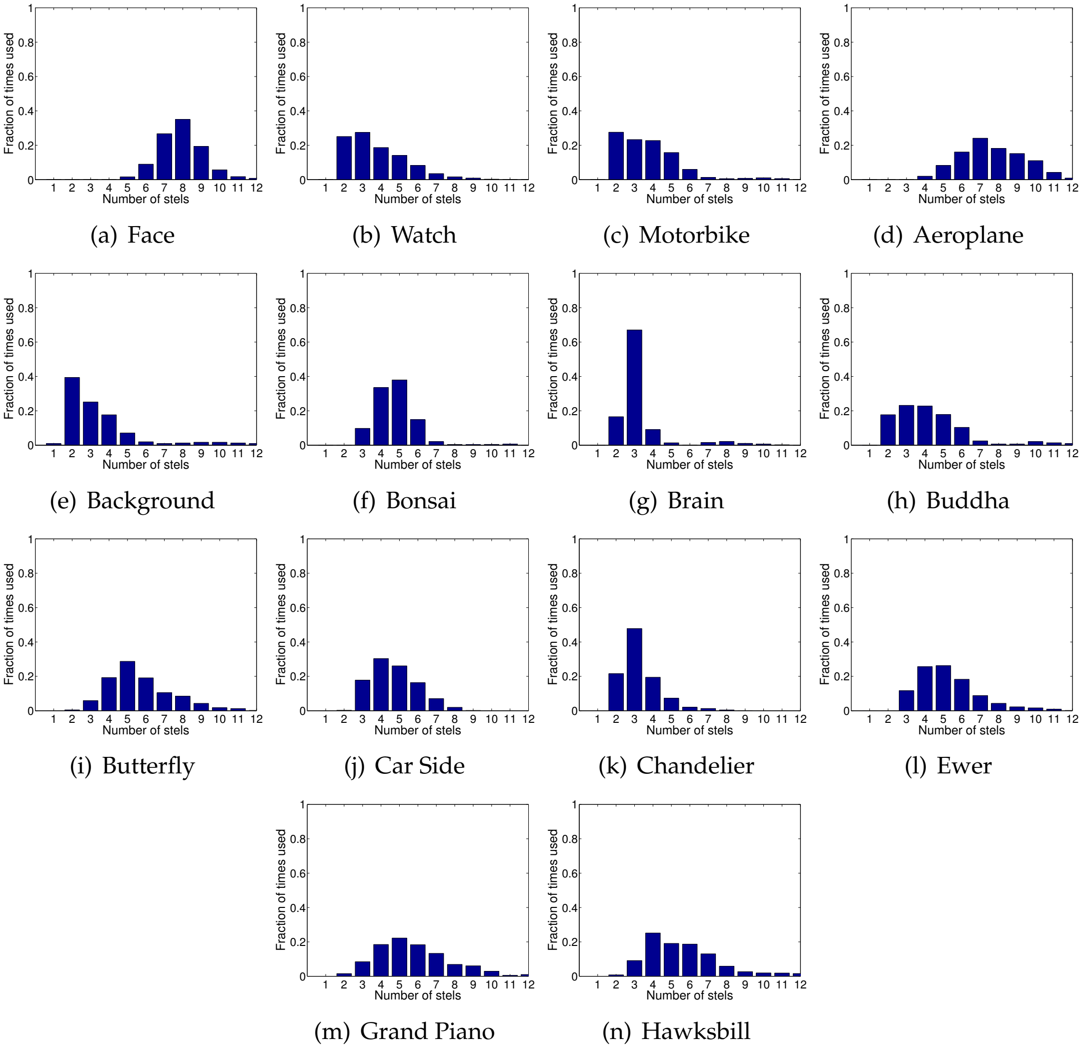
(k) Trilobite

(l) Scorpion

(m) Starfish

(n) Sunflower

Figure 9: Continuation of Fig. 8.



(a) Face (b) Watch (c) Motorbike (d) Aeroplane

(e) Background (f) Bonsai (g) Brain (h) Buddha

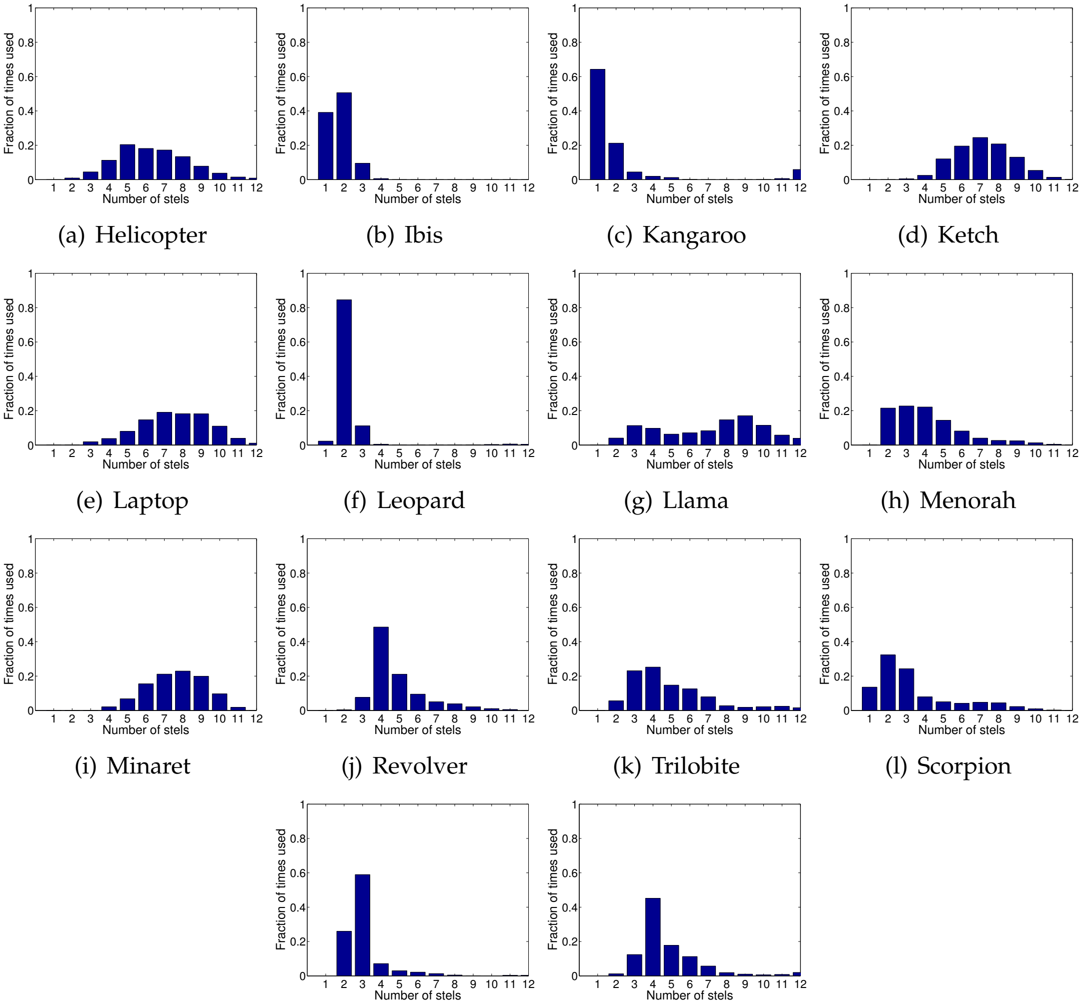
(i) Butterfly (j) Car Side (k) Chandelier (l) Ewer

(m) Grand Piano (n) Hawksbill

Figure 10: Per-class histograms of the number of stels used for all images and all trials.

Figs. 8 and 9 show stel probability maps extracted for every object class, using one of the random trials. We examine in more detail three representative cases: the face (Fig. 8(a)), watch (Fig. 8(b)), and motorbike (Fig. 8(c)) classes. As unused stels (as determined by the aforementioned thresholding) are not shown, it is clear that different objects are using different numbers of stels. One explanation for this may be that different classes have different intrinsic complexities in the parts and colours that comprise them. Alternatively, image classes that have many different poses, deformations, or background clutter may require more stels. Faces appear to be an example of this, since different stels are used to account for variations in expression, hair style and background.

For additional insight, we also examine the per-class histograms over the number of used stels, shown in Figs. 10 and 11. These histograms show the total number of above-threshold stels represented by all images in the class. The histograms were computed by aggregating all post burn-in samples from the Markov chain, across all images in the class and for all trials. The differences between the results illuminate the flexibility of the CIA approach. In some classes, such as watch, the images are effectively modelled using only about four stels, while in other classes, such as face, the images typically require about eight stels.



(a) Helicopter (b) Ibis (c) Kangaroo (d) Ketch

(e) Laptop (f) Leopard (g) Llama (h) Menorah

(i) Minaret (j) Revolver (k) Trilobite (l) Scorpion

(m) Starfish (n) Sunflower

Figure 11: Continuation of Fig. 10.

4.3 Object Recognition Using CIA

In this section, we investigate whether our proposed method (CIA) can be used to extract useful representations for object recognition. For each of the 28 classes, we calculated the training-by-training, and training-by-test Gram matrices, based on the descriptors introduced in Sect. 3, and using the intersection kernel as defined in Eq. (20). Since there are 420 training images, and 420 test images in total, (28 classes with 15 images per class), each of these matrices is a 420×420 matrix. We trained a one-versus-all SVM classifier (Rifkin and Klautau, 2004) using the 28 Gram matrices. These experiments used publicly-available code from (Lazebnik et al., 2006; Chang and Lin, 2001).

One motivation for our Bayesian method is to enable flexibility in the effective number of stels that are learnt. We hypothesized that this flexibility could lead to better representations and thus better performance on classification tasks. Our first comparison is therefore against the non-Bayesian counterpart for CIA, namely the basic non-Bayesian stel model (Jojic and Caspi, 2004). For the basic stel model, it is necessary to set the number of stels in advance. This was done using leave-one-out cross-validation on the training set, allowing for between four and twelve stels. The number of stels was determined by the configuration which produced the best likelihood on held-out validation data.

Overall, the basic stel model with cross-validation obtained 42% classification accuracy, while our Bayesian method obtained 68% accuracy (see Table 1). Thus, we have strong evidence to support the conclusion that the additional flexibility leads to representations that are better for discrimination.

Next, we augmented the descriptor extracted using the Bayesian stel model with a standard two level spatial pyramid descriptor, to see if the combined descriptor would give improved performance. Including the spatial pyramid descriptor yields a modest improvement, increasing the classification accuracy to 72%. While the significance of this improvement is questionable (the standard deviation is about 2%), it may be explained by some classes not being very well modelled using the stel model. Some classes, such as the background_Google and leopard classes, are not well-segmented based on colour and shape co-occurrence. This prevents the stel features from providing useful information. In this case, collecting image features in the scheme of (Lazebnik et al., 2006) provides the classifier with additional information that can be used to improve performance. One way to view collecting image features over a two-level spatial pyramid is that it augments our approach to fine-tune performance on segmentation-unfriendly classes.

Finally, we ask whether our descriptors are comparable to the descriptors typically used for object recognition, based on spatial pyramids. We used the SIFT descriptors over the image and image quadrants (the same representation we appended to our descriptor to obtain the augmented descriptor described above) in conjunction with the intersection kernel. The achieved accuracy was 71%, which is not statistically-significantly different from the result obtained using our method.

In summary, descriptors derived using our proposed method significantly improve on the basic stel model. In particular, our method discovers segmentations of object classes that are more conducive to object recognition. Additionally, our method is competitive with standard SIFT-based spatial pyramid methods, and provides a very different approach to defining feature vectors.

4.4 Effect of Hyperparameter Choices on Learnt Stels

An important question in the context of Bayesian inference is the sensitivity of inferred models to the hyperparameter settings. While it is possible to infer or sample them, we can obtain

Method	Accuracy (sd $\sim$ 2%)
Bayesian stel model (CIA)	68%
Bayesian stel model with spatial pyramid	**72%**
Basic stel model with cross-validated numbers of stels	42%
SIFT with spatial pyramid	71%

Table 1: Classification results on a subset of 28 classes from Caltech101. The first two entries were obtained using our proposed Bayesian method (CIA). The third entry follows the method of (Jojic et al., 2009b), using a fixed number of stels per class that was chosen using cross-validation. The fourth method is considered to be the state of the art and follows (Lazebnik et al., 2006).

intuition and a better understanding of the model by considering how the inferred model changes as a function of the hyperparameter settings.

In this section we report some of our findings when trying different parameter settings for the greyscale version of the model. As the images are greyscale with values in $[0, 1]$, the hyperparameter value for the mean, μ_0, was not expected to be particularly sensitive, and we set it throughout all experiments to be $\mu_0 = 0.5$. However, it was expected that the other normal-inverse Wishart parameters would have an effect on the results. In the following experiments, we centred the hyperparameters at a “reasonable” configuration and then perturbed them one at a time and to examine the effects. The centre values are $\Lambda_0 = 0.001$, $\mu_0 = 0.5$, $\kappa_0 = 0.1$, $\nu_0 = 3$. We examine the results of perturbing κ_0, ν_0, and Λ_0.

4.4.1 Variation in κ_0

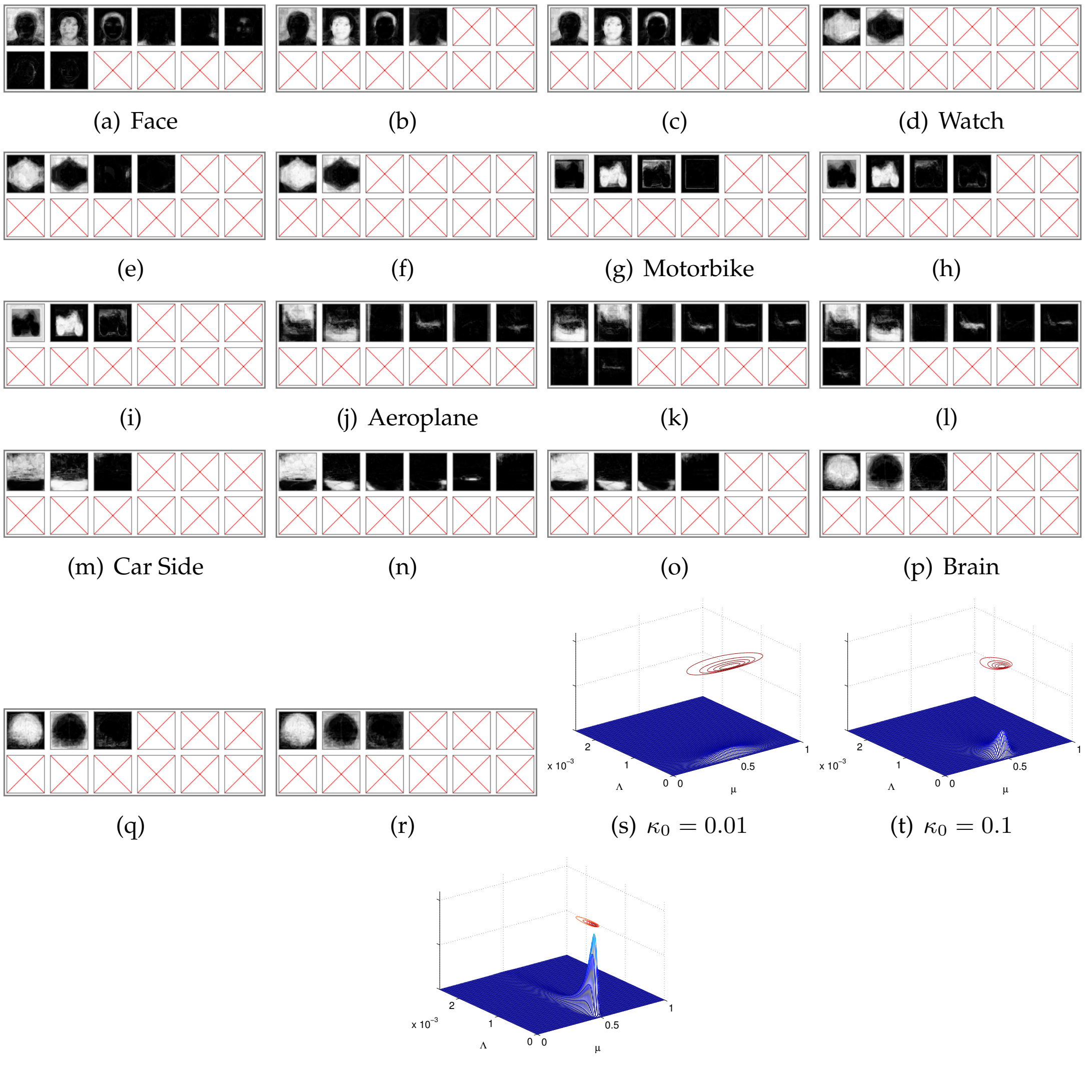
In this set of experiments we considered the values $\kappa_0 \in \{0.01, 0.1, 1\}$. Results are shown in Fig. 12. We observe that for $\kappa_0 = 1$, we obtain many similar stels that tend to be below the threshold. As we lower κ_0, we allow larger variations from the prior mean. For represented stels with many pixels, however, the prior mean has little effect; κ_0 primarily changes the default colour distributions for below-threshold stels. Figs. 12(s)-12(u) show contour plots for the normal-inverse Wishart prior distribution on the mean and variance parameter of the colour model, for each setting of κ_0.

4.4.2 Variation in ν_0

This set of experiments considered the values $\nu_0 \in \{2, 3, 5\}$. Results are shown in Fig. 13. Increasing ν_0 increases the effect of the prior covariance Λ_0. Given the prior choice of Λ_0, this results in narrower colour distributions and decreased flexibility in what pixels each stel can explain. For example, in the faces shown in Fig. 13(c), a single stel cannot explain both the eyes and the face. Figs. 13(s)-13(u), show contour plots for the normal-inverse Wishart prior distribution on the mean and variance parameters of the colour model, as ν_0 is varied.

4.4.3 Variation in Λ_0

These experiments considered the values $\Lambda \in \{0.0001, 0.001, 0.01, \}$. Results are shown in Fig. 14. This hyperparameter corresponds to the prior on the covariance. Although this has a large effect on the normal-inverse-Wishart prior, there is little sensitivity evident in the learnt



(a) Face (b) (c) (d) Watch

(e) (f) (g) Motorbike (h)

(i) (j) Aeroplane (k) (l)

(m) Car Side (n) (o) (p) Brain

(q) (r) (s) $\kappa_0 = 0.01$ (t) $\kappa_0 = 0.1$

(u) $\kappa_0 = 1$

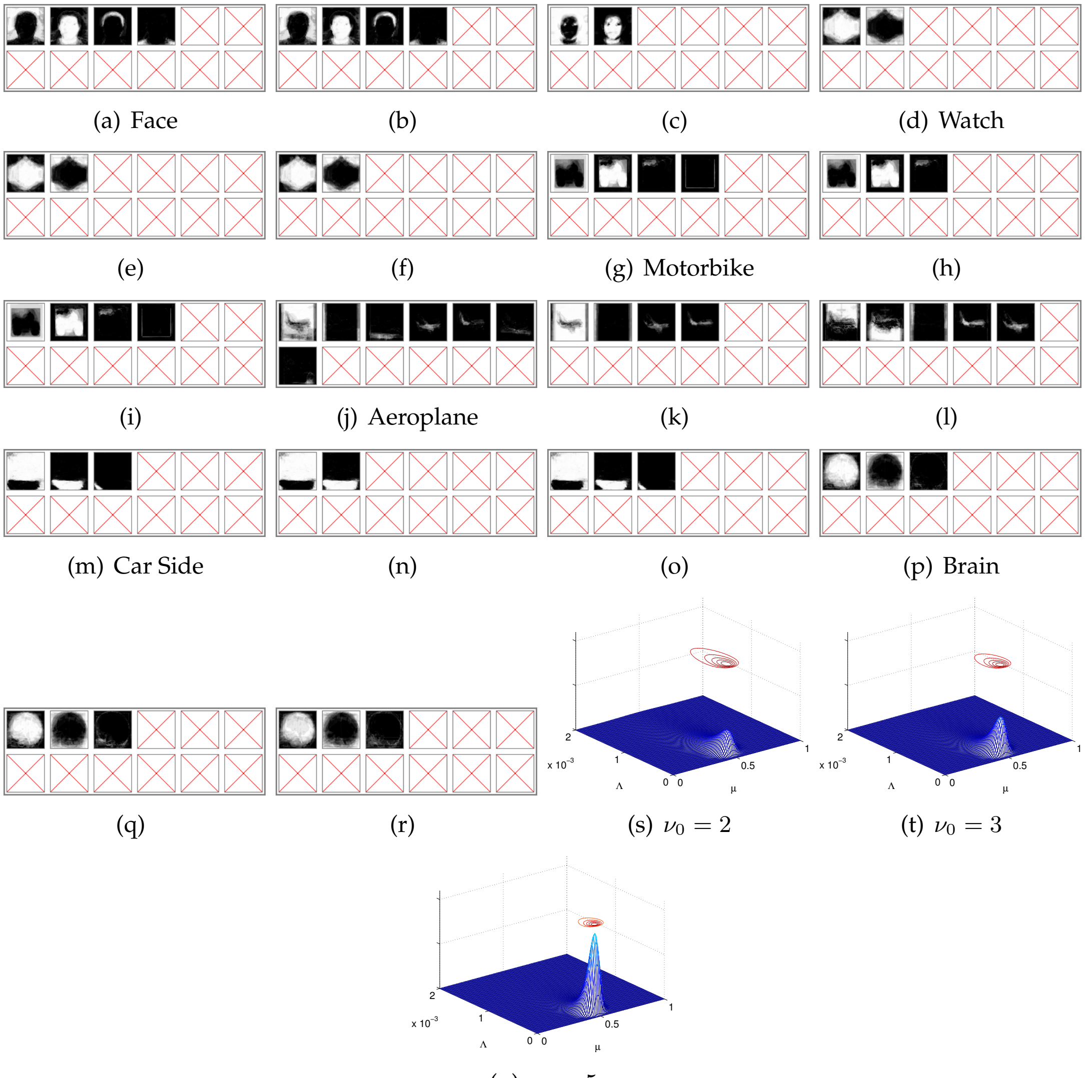
Figure 12: Inferred stels for different settings of κ_0 across columns. The other NIW parameters are set to $\{\mu_0 = 0.5, \Lambda_0 = 0.001, \nu_0 = 3\}$

stels. Figs.14(s)-14(s) show contour plots for the normal-inverse Wishart prior distribution on the mean and variance parameters of the colour model as Λ_0 is varied.

5 Summary

We have introduced a novel generative Bayesian framework, the colour-invariant admixture model (CIA), for generalizing stel models. This rectifies a previous weakness in stel-based models, in which it is difficult to regularize the distribution over stels. Our admixture-based approach represents the full posterior distribution over stel parameters, enabling different numbers of stels to be represented to capture variation in object complexity. We have also introduced a straightforward Gibbs sampler for performing inference in this model. Our empirical analyses demonstrate that CIA is capable of learning varying complexity in stels for each class. Additionally, CIA outperforms stel modelling approaches that require cross-validation, and performs comparably to the popular SIFT-based pyramid matching.

CIA can be used to segment images in an unsupervised fashion, and we have shown that



(a) Face (b) (c) (d) Watch

(e) (f) (g) Motorbike (h)

(i) (j) Aeroplane (k) (l)

(m) Car Side (n) (o) (p) Brain

(q) (r) (s) $\nu_0 = 2$ (t) $\nu_0 = 3$

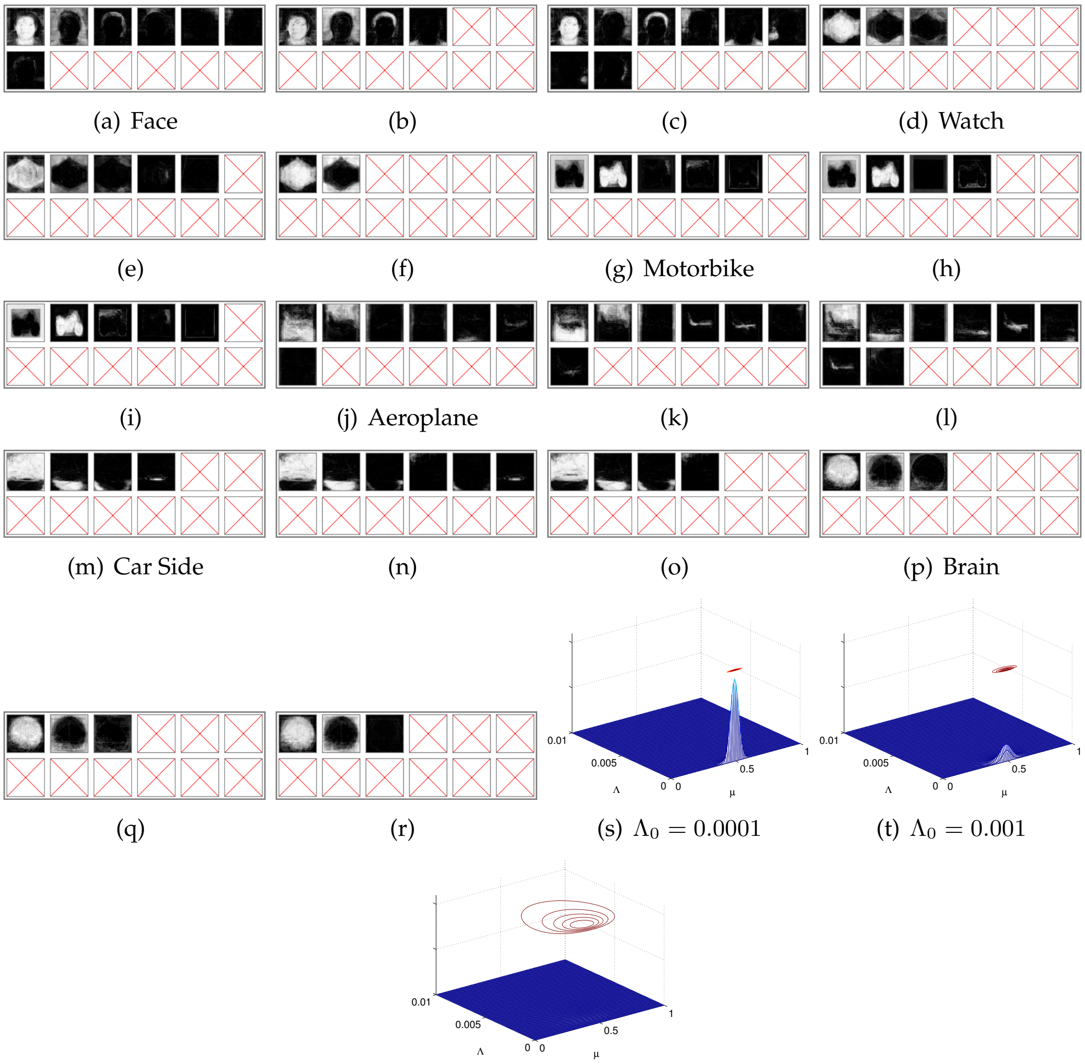
(u) $\nu_0 = 5$

Figure 13: Inferred stels for different settings of ν_0 across columns. The other NIW parameters are set to $\{\mu_0 = 0.5, \Lambda_0 = 0.001, \kappa_0 = 0.1\}$

this segmentation provides a rich backdrop on which to perform object recognition. It robustly captures spatial relations between features, a crucial piece of any foundation for state-of-the-art object recognition.

Although our approach provides a way of inferring a class-specific distribution over the stels, our approach may still be sensitive the maximum number of stels S. For instance, allowing a large number of stels to be used may result in "noisy" stels, in which many stels are speckly and do not appear to represent meaningful structure. This free parameter is a difficult-to-avoid side-effect of this generative model. One future direction for resolving this difficulty is to use a Bayesian nonparametric approach in which an unbounded number of stels are allowed by the model, using, for example, the hierarchical Dirichlet process (Teh et al., 2006).

In addition, patch-based models, such as convolutional neural networks, have recently become popular in the vision community. An extension to this work is to model an image as a collection of smaller patches, all of which use the same palette to colour the image. Here, a vocabulary of patches, similar to Gabor filters, could be learnt, and the types of patches used in an image could be used to discriminate one class from another. For example, one patch


(a) Face (b) (c) (d) Watch

(e) (f) (g) Motorbike (h)

(i) (j) Aeroplane (k) (l)

(m) Car Side (n) (o) (p) Brain

(q) (r) (s) $\Lambda_0 = 0.0001$ (t) $\Lambda_0 = 0.001$

(u) $\Lambda_0 = 0.01$

Figure 14: Inferred stels for different settings of Λ_0 across columns. The other NIW parameters are set to $\{\mu_0 = 0.5, \kappa_0 =, \nu_0 = 3\}$

from the vocabulary may represent the wheel of a motorbike, and so this type of patch being present twice in an image could help distinguish between motorbikes and sunflowers, which have no wheels.

References

D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 8(6):679–698, 1986.

L. Cao and F.-F. Li. Spatially coherent latent topic model for concurrent object segmentation and classification. In *Proceedings of the Eleventh IEEE International Conference on Computer Vision*, 2007.

C. Chang and C. Lin. *LIBSVM: a library for support vector machines*, 2001. `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

D. Erhan, Y. Bengio, A. Courville, P.-A. Manzagol, P. Vincent, and S. Bengio. Why does unsupervised pre-training help deep learning? *Journal of Machine Learning Research*, 11:625–660, 2010.

B. J. Frey and N. Jojic. Estimating mixture models of images and inferring spatial transformations using the EM algorithm. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 416–422, 1999.

B. J. Frey and N. Jojic. Transformation-invariant clustering using the EM algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(1), 2003.

K. Grauman and T. Darrell. The pyramid match kernel: Discriminative classification with sets of image features. In *Proceedings of the Tenth IEEE International Conference on Computer Vision*, pages 1458–1465, 2005.

U. Grenander. *Lectures in Pattern Theory I, II and III: Pattern Analysis, Pattern Synthesis and Regular Structures*. Springer-Verlag, Berlin, 1976–1981.

G. E. Hinton. Connectionist learning procedures. *Artificial Intelligence*, 40:185–234, 1989.

G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

N. Jojic and Y. Caspi. Capturing image structure with probabilistic index maps. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 212–219, 2004.

N. Jojic and B. J. Frey. Learning flexible sprites in video layers. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2001.

N. Jojic, A. Perina, M. Cristani, V. Murino, and B. J. Frey. Stel component analysis: Modeling spatial correlations in image class structure. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 2044–2051, 2009a.

N. Jojic, A. Perina, M. Cristani, V. Murino, and B. J. Frey. Stel component analysis: Modeling spatial correlations in image class structure. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2044–2051, 2009b.

S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2169–2178, 2006.

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, November 1998.

F.-F. Li, R. Fergus, and P. Perona. Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories. In *Proceedings of the IEEE CVPR Workshop on Generative Model Based Vision*, page 178, 2004.

D. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, 1999.

D. Marr. *Vision: A computational investigation into human representation and processing of visual information*. W. H. Freeman and Company, San Franciso, 1982.

D. Mumford. Neuronal architectures for pattern-theoretic problems. In C. Koch and J. Davis, editors, *Large-Scale Theories of the Cortex*, pages 125–152. MIT Press, Cambridge MA., 1994.

K. P. Murphy. Conjugate Bayesian analysis of the Gaussian distribution. Technical report, University of British Columbia, 2007.

B. A. Olshausen and D. J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381:607–609, 1996.

A. Perina, N. Jojic, U. Castellani, M. Cristani, and V. Murino. Object recognition with hierarchical stel models. In *Proceedings of the Eleventh European Conference on Computer Vision*, pages 15–28, 2010.

R. Rifkin and A. Klautau. In defense of one-vs-all classification. *Journal of Machine Learning Research*, 5:101–141, 2004.

B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman. Labelme: a database and web-based tool for image annotation. *International Journal of Computer Vision*, 77:157–173, 2008.

T. Serre, A. Oliva, and T. Poggio. A feedforward architecture accounts for rapid categorization. *Proceedings of the National Academy of Sciences*, 104(15):6424–6429, 2007.

J. Sivic, B. C. Russell, A. A. Efros, A. Zisserman, and W. T. Freeman. Discovering objects and their locations in images. In *Proceedings of the Tenth IEEE International Conference on Computer Vision*, 2005.

E. B. Sudderth, A. Torralba, W. T. Freeman, and A. S. Willsky. Describing visual scenes using transformed object parts. *International Journal of Computer Vision*, 77, 2008.

Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581, 2006.

S. C. Zhu and D. Mumford. GRADE: Gibbs reaction and diffusion equations — a framework for pattern synthesis, image denoising, and removing clutter. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1998.