Deep Part-Based Generative Shape Model with Latent Variables

Alexander Kirillov[1]
alexander.kirillov@tu-dresden

Mikhail Gavrikov[2]
gavrmike@gmail.com

Ekaterina Lobacheva[4]
elobacheva@hse.ru

Anton Osokin[3]
anton.osokin@inria.fr

Dmitry Vetrov[4]
vetrovd@yandex.ru

[1] TU Dresden,
Dresden, Germany

[2] Rubbles,
Moscow, Russia

[3] INRIA – École Normale Supérieure,
Paris, France

[4] National Research University
Higher School of Economics (HSE),
Moscow, Russia

Abstract

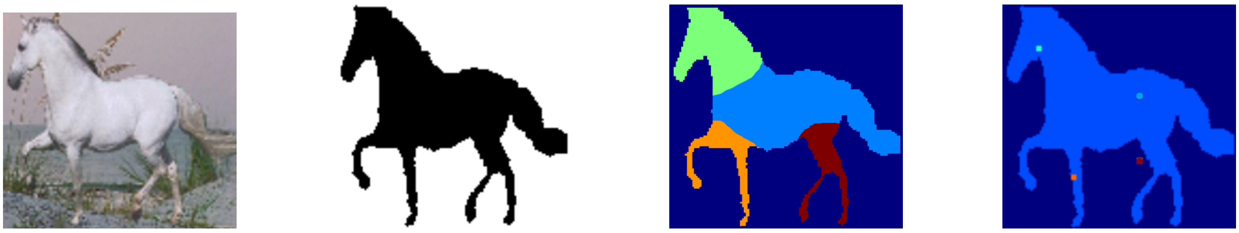
The Shape Boltzmann Machine (SBM) [6] and its multilabel version MSBM [5] have been recently introduced as deep generative models that capture the variations of an object shape. While being more flexible MSBM requires datasets with labeled parts of the objects for training. In the paper we present an algorithm for training MSBM using binary masks of objects and the seeds which approximately correspond to the locations of objects parts. The latter can be obtained from part-based detectors in an unsupervised manner. We derive a latent variable model and an EM-like training procedure for adjusting the weights of MSBM using a deep learning framework. We show that the model trained by our method outperforms SBM in the tasks related to binary shapes and is very close to the original MSBM in terms of quality of multilabel shapes.

1 Introduction

Models of shape play substantial role in a number of computer vision tasks such as segmentation [2, 18], inpainting [3] and detection [10]. Specifically in the case of segmentation, shape models allow to incorporate the prior knowledge about the shape into the segmentation algorithm. Combining low-level features, e.g. color, texture, location, with shape models has been shown to improve the segmentation results [4, 5, 12, 13, 17, 19].

There are many approaches that can capture shape information: active shape models [18], level sets [14], skeleton models [20], grammar models [9], etc. Most of the existing methods either are not flexible enough, or require sophisticated manual labeling making training on large sets of images impossible. The Shape Boltzmann Machine (SBM) offers an elegant way of learning shape models using deep networks of specific type [6]. It allows to generate new shapes which are quite similar (but not identical) to the ones which have been shown to SBM during the training stage. SBM model was later generalized to multilabel case when

Pages 88.1-88.12
DOI: https://dx.doi.org/10.5244/C.30.88



(a) (b) (c) (d)

Figure 1: Image (a) shows a cropped image of an object, (b) – the ground-truth binary segmentation $\boldsymbol{b}$, (c) – the multilabel segmentation $\boldsymbol{m}$, (d) seeds $\boldsymbol{s}$ for the selected 4 parts: head, front legs, rear legs and croup.

the object consists of several labels each responsible for the specific part of the object [5]. Such a model is more expressive since the variations of object's parts are usually easier to learn. The Multinomial SBM (MSBM) has been shown to capture shape properties better but its applicability is limited because it requires annotating all objects parts in addition to binary segmentation masks required for the SBM. Recently a number of segmentation frameworks that use the shape models have been proposed [5, 12, 19].

In this paper we present a method to train an MSBM using only binary masks of the objects together with the seeds of the objects parts. To train an MSBM we establish a latent variable model and an EM-like procedure to optimize the incomplete likelihood. The seeds can be either set manually or found automatically using part-based detectors [9]. Together with the part-based detector (only bounding box annotations required for training) our framework provides the ability to exploit the benefits of multi-part shape models having only the dataset of images with binary object masks annotated.

The main contributions of the paper are the following:

- A joint probabilistic model of a binary mask, a multilabel mask, and object seeds;
- A training procedure allowing to train a multilabel model given only binary mask and seeds that can be obtained automatically.

Our training method requires less annotation than the original one of [5]. We compare the models trained by our technique on two datasets against several available baselines and show that it ties with the MSBM trained by the original method and outperforms the SBM in terms of expressive power measured by shape completion scores proposed in [6]. In addition we show that our approach significantly outperforms several straight-forward algorithms based on automatic generation of multilabel annotations given binary ones and objects seeds.

The rest of the paper is organized as follows. In section 2 we introduce our notation, in section 3 we review the SBM and the MSBM models. In section 4 we present our joint model, and in section 5 – the EM-algorithm to train it. In section 7 we perform the experimental evaluation and conclude in section 8.

2 Notation

Consider a collection of D images of objects of one category normalized to equal resolution, fig. 1a. Let $B = \{\boldsymbol{b}^d\}_{d=1,\dots,D}$ be a set of the binary masks of the objects, fig. 1b, where mask $\boldsymbol{b}^d$ of image d is a binary vector, i.e. elements b_i^d belong to set $\{0,1\}$. Let $M = \{\boldsymbol{m}^d\}_{d=1,\dots,D}$ be a set of multilabel segmentations of the same objects, fig. 1c. Vector $\boldsymbol{m}^d$ contains a label $m_i^d \in 0,\dots,P$ for each pixel i. Here P is the total number of parts of each

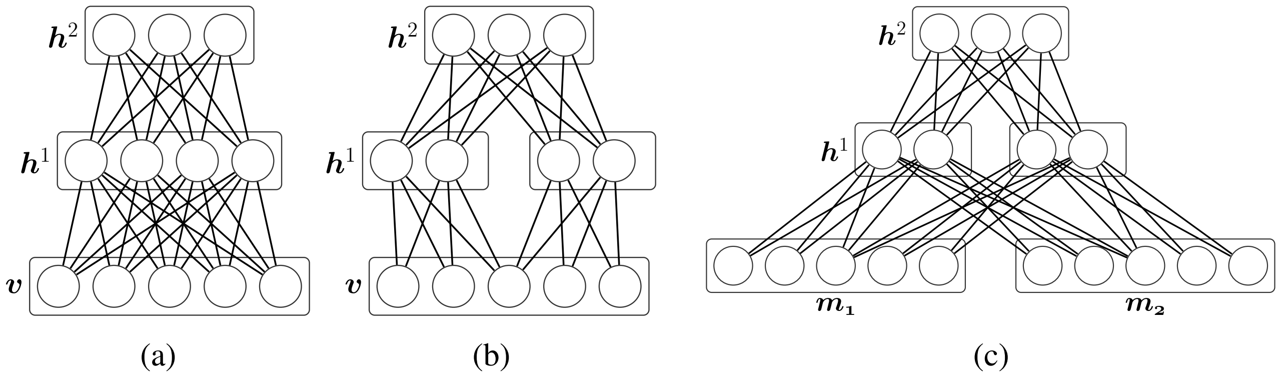


Figure 2: Architectures of deep neural networks: (a) – Deep Boltzmann Machine with 2 hidden layers, (b) – 1D SBM with 2 patches, (c) – 1D MSBM with 2 patches and 2 object parts.

object and equality $m_i^d = p$ indicates that pixel i belongs to part p and $m_i^d = 0$ indicates that pixel i belongs to the background. We use symbol $m_{i,p}^d$ to denote the binary variable indicating that variable m_i^d equals label p, i.e. $m_{i,p}^d = [m_i^d = p]$.[1] In what follows we will omit index d where it is unambiguous from the context.

Let function $f_{coord}(i)$ return the position (x_i, y_i) of pixel i on the image. We also define a set of seeds $S = \{\boldsymbol{s}^d\}_{d=1,\ldots,D}$ where $\boldsymbol{s}^d$ is a vector of seeds $(s_1^d, \ldots, s_P^d)$, one per object's part. Each seed s_p^d, $p = 1, \ldots, P$ is a point in the $2D$ coordinate space (see fig. 1d).

3 Deep-learned shape models

Shape Boltzmann Machine (SBM) [6] is a generative model of binary shapes. The model is a Deep Boltzmann Machine (DBM) [11] with special constraints. The image plane is divided into 4 same-size patches with small overlapping. Each patch is connected with its own set of variables of the first hidden layer. Moreover, weights of these connections have to be equal for each patch. This constraint reduces the number of parameters of deep neural network and allows to avoid overfitting for small datasets and to speed up training procedure. The authors show that such a model stays powerful and has the ability to generate adequate samples of shape. Architectures of DBM and SBM can be seen in fig. 2.

To be more precise, an SBM with 3 layers $\boldsymbol{b}$, $\boldsymbol{h}^1$ and $\boldsymbol{h}^2$ is defined by Gibbs distribution

$$p(\boldsymbol{b}, \boldsymbol{h}^1, \boldsymbol{h}^2 \mid \boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta})} \exp\left(-E(\boldsymbol{b}, \boldsymbol{h}^1, \boldsymbol{h}^2 \mid \boldsymbol{\theta})\right), \tag{1}$$

where $Z(\boldsymbol{\theta}) = \sum_{\boldsymbol{b}, \boldsymbol{h}^1, \boldsymbol{h}^2} \exp\left(-E(\boldsymbol{b}, \boldsymbol{h}^1, \boldsymbol{h}^2 \mid \boldsymbol{\theta})\right)$ is the normalization constant and

$$E(\boldsymbol{b}, \boldsymbol{h}^1, \boldsymbol{h}^2 \mid \theta) = \sum_{i=1}^{I} a_i b_i + \sum_{i=1}^{I} \sum_{j=1}^{J} b_i W_{i,j}^1 h_j^1 + \sum_{j=1}^{J} c_j^1 h_j^1 + \sum_{j=1}^{J} \sum_{k=1}^{K} h_j^1 W_{j,k}^2 h_k^2 + \sum_{k=1}^{K} c_k^2 h_k^2 \tag{2}$$

is the energy function. Vector $\boldsymbol{\theta} = (\boldsymbol{a}, \boldsymbol{c}^1, \boldsymbol{c}^2, W^1, W^2)$ is a concatenation of all the SBM parameters. The constraints that make SBM different from a general DBM are encoded into matrix W^1.

A Multinomial Shape Boltzmann Machine (MSBM) [5] is a generalization of SBM to the task of multilabel segmentation (specifically, a model of an object with certain parts). A 2-layer MSBM with observed layer $\boldsymbol{m}$ and hidden layers $\boldsymbol{h}^1$, $\boldsymbol{h}^2$ is defined using the following Gibbs distribution:

$$p(\boldsymbol{m}, \boldsymbol{h}^1, \boldsymbol{h}^2 \mid \boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta})} \exp\left(-E(\boldsymbol{m}, \boldsymbol{h}^1, \boldsymbol{h}^2 \mid \boldsymbol{\theta})\right), \tag{3}$$

[1] Here $[\cdot]$ is the Iverson bracket notation, i.e. for a predicate A expression $[A]$ equals 1 if A is true and 0 otherwise.

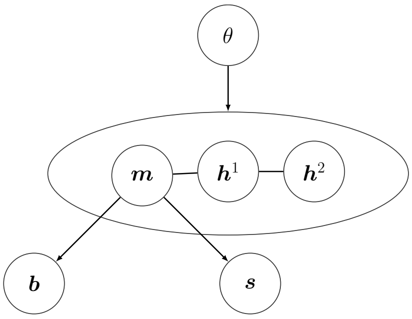


Figure 3: The graphical model illustrating our assumptions on conditional independence of the variables. In this work the model corresponds to the following equation: $p(\boldsymbol{b},\boldsymbol{s},\boldsymbol{m},\boldsymbol{h}^1,\boldsymbol{h}^2|\boldsymbol{\theta}) = p(\boldsymbol{b}|\boldsymbol{m})p(\boldsymbol{s}|\boldsymbol{m})p(\boldsymbol{m},\boldsymbol{h}^1,\boldsymbol{h}^2|\boldsymbol{\theta})$

where $Z(\boldsymbol{\theta}) = \sum_{\boldsymbol{m},\boldsymbol{h}^1,\boldsymbol{h}^2} \exp\left(-E(\boldsymbol{m},\boldsymbol{h}^1,\boldsymbol{h}^2 \mid \boldsymbol{\theta}\right)$ is the normalization constant and

$$E(\boldsymbol{m},\boldsymbol{h}^1,\boldsymbol{h}^2 \mid \boldsymbol{\theta}) = \sum_{i=1}^{I}\sum_{p=1}^{P} a_{i,p}m_{i,p} + \sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{p=1}^{P} m_{i,p}W^1_{i,j,p}h^1_j + \sum_{j=1}^{J} c^1_j h^1_j + \sum_{j=1}^{J}\sum_{k=1}^{K} h^1_j W^2_{j,k} h^2_k + \sum_{k=1}^{K} c^2_k h^2_k, \qquad (4)$$

is the energy function, where $\boldsymbol{\theta} = (a, c^1, c^2, W^1, W^2)$ is a vector of all the MSBM parameters. Parameters W_1 are of specialized structure similarly to the case of SBM.

A state-of-the-art learning procedure for a DBM [15] aims to maximize the log likelihood $\log p(\boldsymbol{b} \mid \boldsymbol{\theta})$ via the EM-algorithm with the fully factorized approximation on the E-step and divides into two phases. The first phase is a layer-wise pretraining, where each layer is trained separately using Stochastic Approximating Procedure (SAP). The second phase is called fine-tuning and consists in optimizing the likelihood w.r.t. all the parameters starting from the result of the first stage.

SBMs and MSBMs are trained in exactly the same way as general DBMs. An important constraint of the scheme is that it requires the full annotation. This is a serious constraint, because obtaining the ground-truth segmentation (especially for multilabel tasks) is not an easy task. In this paper we make a step towards the unsupervised training of a shape model. Specifically, we propose a way to train a multilabel model without using the full multilabel annotation (as in [5]). Instead we use easier-to-obtain binary masks and seed points of the parts.

4 The joint model

We model the joint distribution $p(\boldsymbol{b},\boldsymbol{s},\boldsymbol{m},\boldsymbol{h}^1,\boldsymbol{h}^2|\boldsymbol{\theta})$ of binary mask $\boldsymbol{b}$, seeds $\boldsymbol{s}$, multilabel masks $\boldsymbol{m}$ and hidden variables $\boldsymbol{h}^1$, $\boldsymbol{h}^2$ using the assumption that binary segmentation $\boldsymbol{b}$ and seeds $\boldsymbol{s}$ are conditionally independent given multilabel segmentation $\boldsymbol{m}$, i.e.

$$p(\boldsymbol{b},\boldsymbol{s},\boldsymbol{m},\boldsymbol{h}^1,\boldsymbol{h}^2|\boldsymbol{\theta}) = p(\boldsymbol{b}|\boldsymbol{m})p(\boldsymbol{s}|\boldsymbol{m})p(\boldsymbol{m},\boldsymbol{h}^1,\boldsymbol{h}^2|\boldsymbol{\theta}). \qquad (5)$$

Distribution $p(\boldsymbol{m},\boldsymbol{h}^1,\boldsymbol{h}^2|\boldsymbol{\theta})$ is represented by an MSBM Gibbs distribution (3). Further we propose simple assumptions about $p(\boldsymbol{b}|\boldsymbol{m})$ and $p(\boldsymbol{s}|\boldsymbol{m})$ decomposition. Conditional distribution $p(\boldsymbol{b}|\boldsymbol{m})$ decomposes into independent distributions $p(b_i|m_i)$ for each pixel i. We use an intuitive assumption that if a pixel belongs to any part of an object, then it belongs to the object with probability 1, otherwise this pixel belongs to the background. Hence, $p(\boldsymbol{b}|\boldsymbol{m}) = \prod_i p(b_i|m_i)$ where $p(b_i|m_i)$ is a simple degenerate distribution

$$p(b_i|m_i) = [b_i = 0][m_i = 0] + [b_i \neq 0][m_i \neq 0]. \tag{6}$$

W.r.t. $p(\boldsymbol{s}|\boldsymbol{m})$ we assume that each pixel impacts the seed positions independently, and the pixels belonging to the background do not impact the seeds. Moreover, we suppose that pixel i only impacts the position of seed s_{m_i}. Consequently, distribution $p(s|m)$ is proportional to $\prod_{i:m_i \neq 0} p(s_{m_i}|m_i)$. We use the Gaussian distribution to model the factors in this product:

$$p(s|m) \propto \prod_{i:m_i \neq 0} \mathcal{N}\left(s_{m_i}|f_{coord}(i), \sigma^2\right) \propto \prod_{i:m_i \neq 0} \exp\left\{-\frac{\|s_{m_i} - f_{coord}(i)\|_2^2}{2\sigma^2}\right\} \tag{7}$$

where σ is the external parameter of the method. We assume that pixel i belonging to the part of object m_i attracts seed s_{m_i} to be closer to position $f_{coord}(i)$.

5 The EM-algorithm

To train the unknown parameters $\boldsymbol{\theta}$ of the Gibbs distribution used as a part of our model we use the Expectation-Maximization algorithm, i.e. maximize $\log P(B, S \mid \boldsymbol{\theta}) =$ $= \sum_{d=1}^{D} \log p(\boldsymbol{b}^d, \boldsymbol{s}^d|\boldsymbol{\theta})$ w.r.t. parameters $\boldsymbol{\theta}$. Here we assume that variables $\boldsymbol{b}$, $\boldsymbol{s}$ are observed and $\boldsymbol{m}$, $\boldsymbol{h}^1$, $\boldsymbol{h}^2$ are hidden. At the E-step we obtain distributions $q^d(\boldsymbol{m}, \boldsymbol{h}^1, \boldsymbol{h}^2)$ close to the posterior of model (5) on the unobserved variables in the sense of KL-divergence:

$$\min_{q^d} \mathrm{KL}\left(q^d(\boldsymbol{m}, \boldsymbol{h}^1, \boldsymbol{h}^2) \| p(\boldsymbol{m}, \boldsymbol{h}^1, \boldsymbol{h}^2|\boldsymbol{b}^d, \boldsymbol{s}^d, \boldsymbol{\theta})\right). \tag{8}$$

To make this tractable we optimize only in the family of fully-factorized distributions:

$$q^d(\boldsymbol{m}, \boldsymbol{h}^1, \boldsymbol{h}^2) = \prod_{i=1}^{I} q_i^d(m_i) \prod_{j=1}^{J} q_j^d(h_j^1) \prod_{k=1}^{K} q_k^d(h_k^2) \tag{9}$$

During the M-step, we update parameters $\boldsymbol{\theta}$ by solving the following maximization problem:

$$\max_{\theta} \sum_{d=1}^{D} \left[\sum_{\boldsymbol{m}, \boldsymbol{h}^1, \boldsymbol{h}^2} q^d(\boldsymbol{m}, \boldsymbol{h}^1, \boldsymbol{h}^2) \log p(\boldsymbol{b}^d, \boldsymbol{s}^d, \boldsymbol{m}, \boldsymbol{h}^1, \boldsymbol{h}^2|\boldsymbol{\theta}) \right]. \tag{10}$$

Further, we discuss the E- and M-steps in more detail.

The Expectation step. We optimize objective (8) in the fully-factorized family (9) using the standard result from [1, p. 466]:

$$q_z^d(m) \propto \exp\left(\mathbb{E}_{r \neq z} \log p(\boldsymbol{b}^d|\boldsymbol{m}) p(\boldsymbol{s}^d|\boldsymbol{m}) p(\boldsymbol{m}, \boldsymbol{h}^1, \boldsymbol{h}^2|\theta)\right) \tag{11}$$

where index z defines factors from (9) and $\mathbb{E}_{r \neq z}$ is expectation over all factors from (9) except factor z. Using (11) we obtain

$$\hat{q}_i^d(m_i = p) = [b_i^d \neq 0][m_i \neq 0] \exp\left(-\frac{1}{2\sigma^2}\left\|s_{m_i}^d - f_{coord}(i)\right\|_2^2 + a_{i,m_i} + \sum_j W_{i,j,m_i}^1 q_j(h_j^1 = 1)\right) + [b_i^d = 0][m_i^d = 0] \tag{12}$$

where $\hat{q}_i(m_i)$ is not normalized. To compute normalized distribution $q_i(m_i)$ recall that $m_i \in \{0,1,\ldots,P\}$, therefore

$$q_i^d(m_i = p) = \frac{\hat{q}_i^d(m_i = p)}{\sum_{p'} \hat{q}_i^d(m_i = p')}, \qquad p \in \{0,1,\ldots,P\}. \tag{13}$$

Applying results analogous to (11) for other variables gives us equations to recompute $\boldsymbol{h}^1$ and $\boldsymbol{h}^2$:

$$q_j^d(h_j^1 = 1) = \sigma(c_j^1 + \sum_{i,p=1}^{P} q_i(m_i = p)W_{i,j,p}^1 + \sum_k W_{j,k}^2 q_k(h_k^2 = 1)), \tag{14}$$

$$q_k^d(h_k^2 = 1) = \sigma(c_k^2 + \sum_j q_j(h_j^1 = 1)W_{j,k}^2), \tag{15}$$

where $\sigma(x) = \frac{1}{1+e^{-x}}$. According to [1], during variational inference procedure we iterate over (12)-(13), (14), (15). We also developed E-step in a case when some seeds are unknown. The summary of the method is presented in the supplementary material.

Notice that (14) and (15) are almost equal to the variational approximation on the E-step in the classic scheme of DBM learning procedure [16]. The difference is that the default scheme in (14) uses known multilabel segmentation, whereas our approach uses distribution under multilabel segmentation obtaining from (12)-(13).

The Maximization step. On the M-step we maximize (10) given distribution $q(\boldsymbol{m}, \boldsymbol{h}^1, \boldsymbol{h}^2)$ obtained on the E-step. Using (5) we rewrite (10)

$$\max_{\theta} \sum_{d=1}^{D} \Bigg(\sum_{i=1}^{I} \sum_{p=1}^{P} a_{i,p} q_i(m_i^d = p) + \sum_{i=1}^{I} \sum_{j=1}^{J} \sum_{p=1}^{P} q_i(m_i^d = p) W_{i,j,p}^1 q_j(h_j^1 = 1) + \sum_{j=1}^{J} c_j^1 q_j(h_j^1 = 1)$$
$$+ \sum_{j=1}^{J} \sum_{k=1}^{K} q_j(h_j^1 = 1) W_{j,k}^2 q_k(h_k^2 = 1) + \sum_{k=1}^{K} c_k^2 q_k(h_k^2 = 1) - \log Z(\boldsymbol{\theta}) \Bigg). \tag{16}$$

Maximization problem (16) can be solved using stochastic approximation procedure [16].

Note, that the overall procedure has the same computational complexity as the MSBM training procedure [5] where lower bound of $\sum_{d=1}^{D} \log p(\boldsymbol{m}^d|\boldsymbol{\theta})$ is minimized. During this minimization, the variational inference via (13), (14), (15) is used to obtain $\mathbb{E}_{data}\boldsymbol{h}^1$ and $\mathbb{E}_{data}\boldsymbol{h}^2$. In our procedure we compute them on E-step only. Hence, the computational complexities of this two procedures are equal.

6 The usage of a part-based detector

In the previous sections we have described a procedure that allows to train MSBM given only binary ground-truth masks and seeds for all parts of the objects.

If a dataset contains only labelled objects, but not the parts (e.g. Weizmann horses [2], Caltech-101 [8], Pascal VOC [7], etc.) it is only required to annotate the seeds of the object parts. There are at least two ways to set these seeds up: setting the seeds manually and the automated way by using a part-based detector [9]. Setting the seeds manually is a much easier operation than providing the detailed mask for each part, and thus even in this setting our approach can be useful.

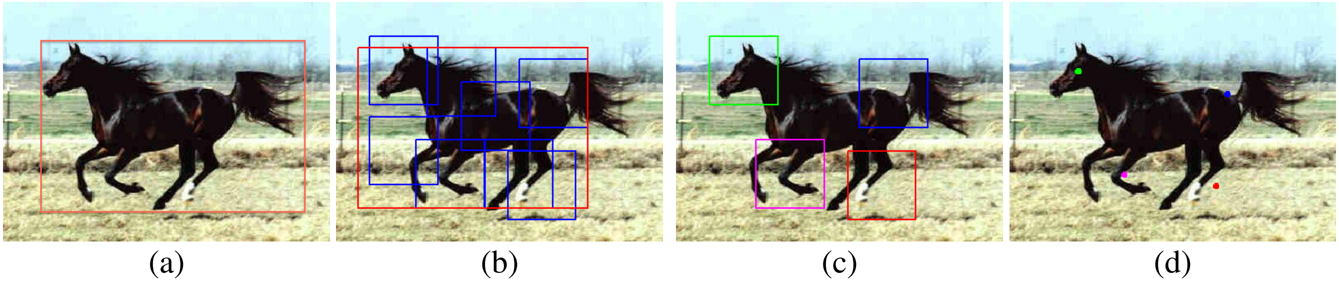


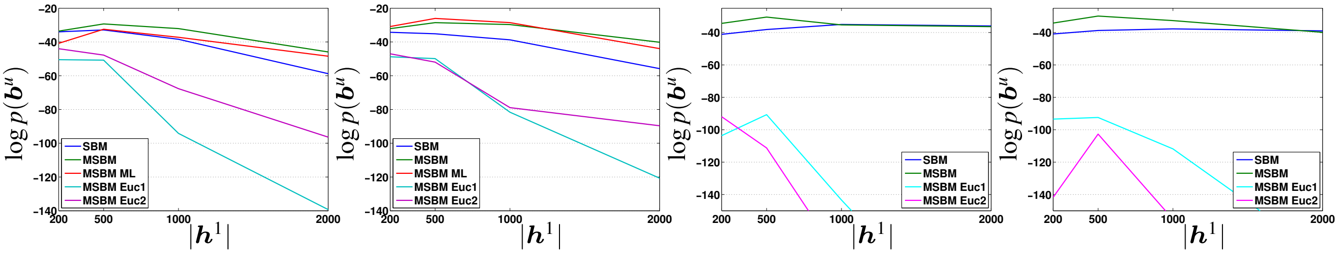
Figure 4: Data examples: (a) image with a bounding box around the object, (b) detector output, main bounding box and bounding boxes for parts, (c) chosen parts, (d) seeds for chosen parts.

An alternative way consists in using a detector that can automatically identify parts that remain fairly consistent in all images. The part-based detector is trained with a dataset of color images annotated by bounding boxes around each object of interest, fig. 4a. Note, that if a dataset contains the binary masks one can easily obtain the required bounding boxes. The detector chooses particular small areas of the image that have similar structures across the majority of the dataset images and defines them as certain "parts" of the image. During the detection phase, the detector finds the parts and then selects the ones fitting the right positioning on the whole image. Thus, we obtain not only the bounding box for the entire object, but also small bounding boxes for each part, fig. 4b. For example, for the horse dataset [2] the detector selects the parts for the head, front legs, rear legs, and the croup. The chosen parts do not always belong to the object and might represent a part of the image as well, especially when the dataset is small. Nevertheless, such occurrences are quite easily detected manually, and the non-object parts are excluded, fig. 4c. We employ detector with 8 parts. After training we choose 4 parts and take the centres of the bounding boxes as the seeds, fig. 4d. The advantages of using the part-based detector is primarily the time- and labor- efficiency of the overall procedure. However, manual seed setting gives us the flexibility of choosing the parts of the object to use, which improves preciseness of the overall MSBM model.

7 Experiments

The authors of [6] and [5] have shown significant evidence of the generalisation abilities and realism of SBM and MSBM as shape models. In this paper we show that an MSBM trained without the full ground-truth annotation (our approach) generates samples similar to an MSBM trained with the annotated parts and significantly outperforms the MSBM models trained using the annotation created by simple baselines. Moreover we show the benefits of using the MSBM trained with our procedure compared to the SBM model in the sense of quality of generated binary shapes.

We perform all the experiments on the two datasets: the Weizmann horse dataset [2] (327 images) and the Caltech-101 motorbikes [8] (798 images). The description of the datasets is provided in the supplementary material. To apply the MSBM method of [5] and for the sake of evaluation we manually created full annotations (with all parts labeled) for the Weizmann horse dataset. The average time required to label one image was about 10 seconds given the specialized graphical interface. For all the experiments we use a desktop machine with `core i7 2.8GHz CPU` and `12GB RAM`. Our `Cython` implementation of our training method requires around 2 hours to train a model with 2000 units on the first hidden layer and 200 units on the second hidden layer on the Weizmann horse dataset.



(a) $|\boldsymbol{h}^2| = 100$ (b) $|\boldsymbol{h}^2| = 200$ (c) $|\boldsymbol{h}^2| = 100$ (d) $|\boldsymbol{h}^2| = 200$

Figure 5: Imputation scores for the different models with different number of units on the hidden layers for Weizmann dataset – (a), (b) and for Caltech-101 motorbikes – (c), (d) (the higher the better). The blue lines correspond to the SBM model [6], the red ones – to the MSBM trained using the fully annotated ground truth, the green ones – trained with our EM-based method, the cyan and the magenta lines – to the MSBM trained using the multilabel segmentations obtained by Euc1 and Euc2, respectively.

Baselines. We use several baselines for the comparisons. Firstly, we use the MSBM model trained using the full multilabel annotations by the original method of [5]. Secondly, we apply the method of [5] on top of some heuristic ways to generate full annotations given binary annotations and seeds of all parts. Specifically we associate each pixel with the part whose seed is closest to the pixel in the image plane. We use two ways to measure the distance: the naive Euclidean distance (Euc1) and the Euclidean length of the shortest path that lies within the object (Euc2). Some examples of generated annotations are shown in the supplementary material. Note that one iteration of our EM-procedure is very similar to the baseline Euc1. The main visual difference between the manual and automatic annotations is that automatic ones have large variations of the inner boundaries between the parts of the objects, while the inner boundaries of the manual ones are more smooth and consistent.

Comparison of MSBMs trained differently. In this experiment we qualitatively compare the MSBM trained with the original method of Eslami et al. [5], the one trained with our technique and the MSBMs trained using automatically obtained multilabel annotations. We train all MSBMs with 1000 units on the first hidden layer and 100 units on the second hidden layer. The size of patch overlap in the model is 4. To train the MSBM with the original method we use the manually-obtained annotations of the parts. To train the MSBM with our EM-based method we use the part-based detector [9] to obtain the seeds for all object parts. The same seeds are used for the annotations based on the closest seeds (Euc1 and Euc2).

Please see the supplementary material for the generated samples. Samples were obtained using the standard MCMC procedure [6] repeated 200 times.

Note that both the MSBM trained with fully annotated dataset and the MSBM trained by our method produce rather smooth outer boundaries, but the boundaries between the parts for our model are more noisy.by

The outer boundaries of the MSBMs trained with automatically obtained multilabel segmentations are very noisy and some shapes do not resemble objects. The probable reason of the failure of this approach is that automatically obtained multilabel segmentations have large variations of the inner boundaries between the parts of the objects through a dataset, while the manual multilabel segmentations have quite predictive patterns in the inner boundaries.

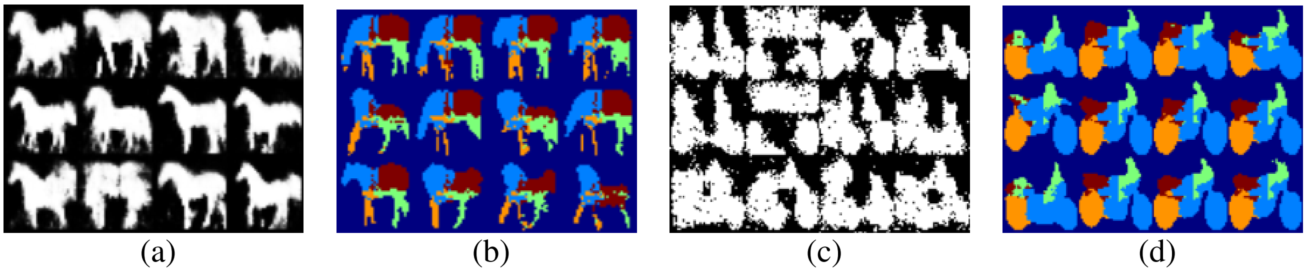


Figure 6: (a), (b) – shapes generated from the seeds for Weizmann dataset: (a) – the SBM samples, (b) – the MSBM samples; (c), (d) – shapes generated from the seeds for Caltech-101 motorbikes: (c) – the SBM samples, (d) – the MSBM samples.

Shape completion. We conduct the "imputation score" experiment (analogous to [6]) to quantitatively compare the MSBMs trained with different procedures: algorithm of [5] using the manual annotation (ML) and the automatically generated annotation (Euc1 and Euc2). We also provide the results of the binary SBM as a baseline. Specifically, we divide each test image into 9 segments (3 by 3 grid). For each test image and each of its segments we estimate the conditional probability of the binary ground-truth mask of the selected segment given the mask of the remaining 8 segments. The log probability is then averaged over all the images and all the segments.

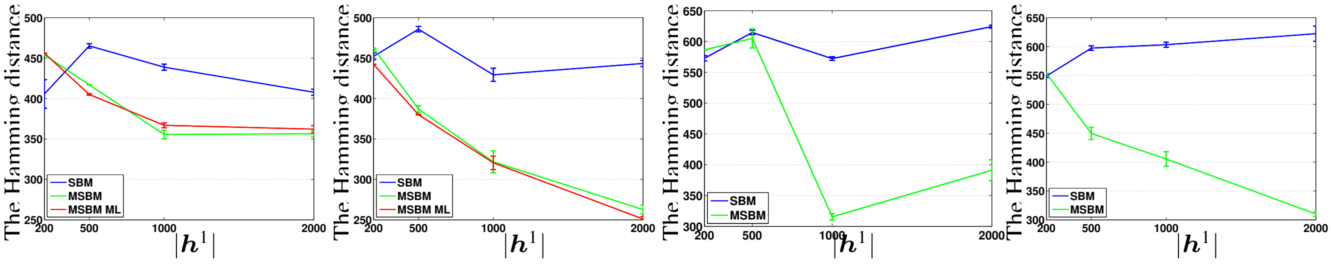
Let $\boldsymbol{b}^o$ be the observed part of the image mask and $\boldsymbol{s}^o$ be the vector of seeds from this part. Let also $\boldsymbol{b}^u$ be currently unobserved 9-th segment and $\boldsymbol{s}^u$ are seeds from it. In case of SBM we estimate $p(\boldsymbol{b}^u|\boldsymbol{b}^o)$, whereas for MSBM we estimate $p(\boldsymbol{b}^u|\boldsymbol{b}^o,\boldsymbol{s}^o)$. Here, $p(\boldsymbol{b}^u|\boldsymbol{b}^o) = \sum_{r=1}^{R} p(\boldsymbol{b}^u|\boldsymbol{h}_r)/R$ and $p(\boldsymbol{b}^u|\boldsymbol{b}^o,\boldsymbol{s}^o) = \sum_{r=1}^{R} p(\boldsymbol{b}^u|\boldsymbol{h}_r)/R$, where $\boldsymbol{h}_r, r = 1,\ldots,R$ are samples obtained from $p(\boldsymbol{h}|\boldsymbol{b}^o)$ for the SBM and $p(\boldsymbol{h}|\boldsymbol{b}^o,\boldsymbol{s}^o)$ for the MSBM via the MCMC procedure. In case $\boldsymbol{s}^u$ is not empty, we also sample it during the MCMC procedure. Please, refer the supplementary material for the equation of seeds' probability given over variables. In the MCMC procedure every time we obtain $\boldsymbol{b}$ we enforce the observed part of shape to be the same as $\boldsymbol{b}^o$.

All the models have the patch overlap equal to 4. For each model each layer was pre-trained for 1000 iterations and the fine-tuning (joint training) was run for 2000 iterations.

Results of the experiment for different architectures of SBM and MSBM are reported in fig. 5. The experiment shows that MSBM trained using automatically obtained full annotation is even worse than the SBM. The MSBM trained with our method (without additional manual annotations) provides the imputation scores comparable to the ones of the MSBM trained using full annotation and the better values of the score compared to the SBM and other models. Please see the supplementary material for more results.

Shape generation from the seeds. In order to show a new feature that the MSBM model has compared to the SBM we experiment with a task of generating the shapes if only seeds are known. Here we assume that seeds $\boldsymbol{s}$ are known and pixels $\boldsymbol{b}^s$ such that $f_{coord}(b^s_p) = s_p, \forall p \in \{1,\ldots,P\}$ belong to the object. We find the hidden variables using $p(\boldsymbol{h} \mid \boldsymbol{b}^s)$ for the SBM and $p(\boldsymbol{h} \mid \boldsymbol{b}^s, \boldsymbol{s})$ for the MSBM via MCMC. After we obtain the hidden variables, we can generate the shape (seed pixels are fixed to belong to the object). The results of both SBM and MSBM after 100 iteration of MCMC given the seeds of the test shapes are shown in fig. 6.

To analyse the result quantitatively we compute the Hamming distance between the test shapes and the shapes which were generated from the seeds. Comparison of the SBM against the MSBM is shown in fig. 7. This experiment shows that the MSBM model of shape can



(a) $|h^2| = 100$ (b) $|h^2| = 200$ (c) $|h^2| = 100$ (d) $|h^2| = 200$

Figure 7: The Hamming distance between the test shapes and shapes generated by SBM, MSBM from the seeds for Weizmann dataset – (a), (b) and for Caltech-101 motorbikes – (c), (d) (the lower the better).

better guess the shape if only seeds are known.

8 Conclusions

In the paper we present a framework for training multinomial shape Boltzmann machine using only binary masks of the objects together with the seeds of objects parts. We show that the latter can be effectively obtained in an automatic manner from the part-based detector. Our EM-based algorithm provides the flexibility of multinomial shape Boltzmann machine, outperforms the binary SBMs in representing the binary shapes and shows almost equal quality as the MSBM trained by original procedure with manual multilabel segmentations. We also find out that the MSBM trained by new procedure significantly outperforms the MSBMs trained with multilabel segmentations obtained by some straight-forward heuristic procedure from the binary segmentations and the seeds. Additionally, our method can be easily extended to the case of missing seeds.

9 Acknowledges

This work was partially supported by the European Research Council (ERC) under the European Unions Horizon 2020 research and innovation programme (grant agreement No 647769), Microsoft Research - INRIA joint centre, RFBR project No. 15-31-20596 (mol-a-ved) and Microsoft: Moscow State University Joint Research Center (RPD 1053945).

References

[1] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006. ISBN 0387310738.

[2] Eran Borenstein. Combining top-down and bottom-up segmentation. In *In Proceedings IEEE workshop on Perceptual Organization in Computer Vision, CVPR*, page 46, 2004.

[3] Tony F. Chan and Jianhong Shen. Non-texture inpainting by curvature-driven diffusions (cdd). *J. Visual Comm. Image Rep*, 12:436–449, 2001.

[4] Fei Chen, Huimin Yu, Roland Hu, and Xunxun Zeng. Deep learning shape priors for object segmentation. In *CVPR*, June 2013.

[5] S. M. Ali Eslami and Chris Williams. A generative model for parts-based object segmentation. In P. Bartlett, F.C.N. Pereira, C.J.C. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 100–107. 2012.

[6] S. M. Ali Eslami, Nicolas Heess, Christopher K. I. Williams, and John Winn. The shape boltzmann machine: a strong model of object shape. In *International Journal of Computer Vision*, 2013.

[7] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes (VOC) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010.

[8] L. Fei-Fei, R. Fergus, and P. Perona. One-shot learning of object categories. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(4):594–611, 2006.

[9] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645, 2010.

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

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

[12] Andrew Kae, Kihyuk Sohn, Honglak Lee, and Erik Learned-Miller. Augmenting crfs with boltzmann machine shape priors for image labeling. In *CVPR*, June 2013.

[13] Yujia Li, Daniel Tarlow, and Richard Zemel. Exploring compositional high order pattern potentials for structured output learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2013.

[14] M. Rousson and N. Paragios. Shape priors for level set representations. In *European Conference on Computer Vision*, volume 2351 of *Lecture Notes in Computer Science*, pages 78–92, 2002.

[15] R. Salakhutdinov and G. Hinton. An efficient learning procedure for deep boltzmann machines. *Neural Computation*, 24(8):1967–2006, 2012.

[16] Ruslan Salakhutdinov and Hugo Larochelle. Efficient learning of deep boltzmann machines. In Yee Whye Teh and D. Mike Titterington, editors, *AISTATS*, volume 9 of *JMLR Proceedings*, pages 693–700. JMLR.org, 2010.

[17] Stavros Tsogkas, Iasonas Kokkinos, George Papandreou, and Andrea Vedaldi. Semantic part segmentation with deep learning. *arXiv preprint arXiv:1505.02438*, 2015.

[18] B. Van Ginneken, A. F. Frangi, J. J. Staal, B. M. ter Haar Romeny, and M. A. Viergever. Active shape model segmentation with optimal features. *IEEE Transactions on Medical Imaging*, 21(8):924–933, 2002.

[19] Jimei Yang, Simon Safar, and Ming-Hsuan Yang. Max-margin boltzmann machines for object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 320–327, 2014.

[20] B. Yangel and D. Vetrov. Image segmentation with a shape prior based on simplified skeleton. In *Energy Minimization Methods in Computer Vision and Pattern Recognition*, volume 6819 of *Lecture Notes in Computer Science*, pages 247–260, 2011.