Semantic Part Segmentation using Compositional Model combining Shape and Appearance

Jianyu Wang, Alan Yuille
University of California, Los Angeles
wjyouch@ucla.edu, yuille@stat.ucla.edu

Abstract

In this paper, we study the problem of semantic part segmentation for animals. This is more challenging than standard object detection, object segmentation and pose estimation tasks because semantic parts of animals often have similar appearance and highly varying shapes. To tackle these challenges, we build a mixture of compositional models to represent the object boundary and the boundaries of semantic parts. And we incorporate edge, appearance, and semantic part cues into the compositional model. Given part-level segmentation annotation, we develop a novel algorithm to learn a mixture of compositional models under various poses and viewpoints for certain animal classes. Furthermore, a linear complexity algorithm is offered for efficient inference of the compositional model using dynamic programming. We evaluate our method for horse and cow using a newly annotated dataset on Pascal VOC 2010 which has pixelwise part labels. Experimental results demonstrate the effectiveness of our method.
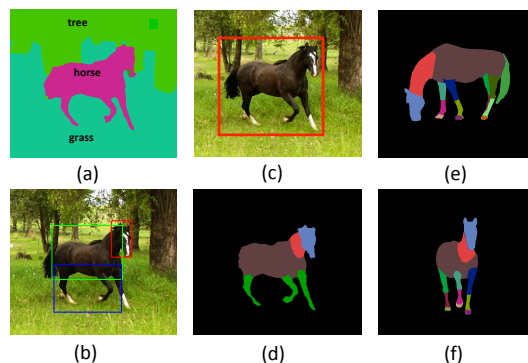
Figure 1: Different visual recognition tasks: (a) semantic labeling with pixelwise object and background label. (b) object detection which outputs bounding box. (c) part detection which gives bounding box for part. (d,e,f) semantic part segmentation with pixelwise part label. We study the semantic part segmentation problem in this paper. Best viewed in color.

1. Introduction

The past few years have witnessed significant progress on various object-level visual recognition tasks, such as object detection [13, 17], object segmentation [5, 1, 7], etc. Understanding how different parts of an object are related and where the parts are located have been an increasingly important topic in computer vision. There is extensive study on some part-level visual recognition tasks, such as human pose estimation (predicting joints) [30, 27] and landmark localization (predicting keypoints) [3, 22]. But there are only a few pieces of works on semantic part segmentation, such as human parsing [4, 10, 9, 29] and car parsing [26, 11, 23]. In some applications (e.g., activity analysis), it would be of great use if computers can output richer part segmentation instead of just giving a set of keypoints/landmarks, a bounding box or an entire object.

We make an attempt on the challenging task of semantic part segmentation for animals in this paper. Since animals often have homogeneous appearance (e.g., furs) on the whole body, mid-level segmentation methods [6, 2] could not output quality proposals for semantic parts. Besides, current classifers are not able to distinguish between different semantic parts since they often have similar appearance. Thus we could not simply take the popular object segmentation pipeline [5, 1] by treating each semantic part as an object. This tells us that the shape information of semantic parts are necessary for part segmentation. But there is a large amount of variability of part shapes due to different animal viewpoints and poses (see (d,e,f) in Figure 1). Therefore, it is very challenging to build a model that effectively combines animal appearance, part shape and spatial relation among parts under varying viewpoints and poses, while still allowing efficient learning and inference.

Inspired by [19, 34, 32, 20], compositional model is able to capture long-range relations among parts while still en-

abling efficient inference since parts are arranged in a hierarchical manner. The intuition of compositional model is that articulated objects are often built by compositions of the parts, which in turn are built by compositions of more elementary subparts. Specifically, in this paper, we build a mixture of compositional models to represent the animal and part shapes/boundaries. Each mixture is able to handle local deformation of shapes and different mixtures deal with global variations due to viewpoints and poses. We incorporate edge, appearance, and part cues into the compositional model by using algorithms from edge detection, semantic labeling and part detection.

It is of significant importance to design efficient *inference* and *learning* algorithms for the compositional model. We develop the constrained generalized distance transform (CGDT) algorithm which extends the distance transform algorithm in [12]. This algorithm allows us to perform efficient linear-time inference for the model. Besides, we design a novel algorithm to learn the compositional models for animal and parts boundaries under various poses and viewpoints from the part-level annotation. And we learn the parameters of the model using latent SVM.

In order to segment highly deformable animal legs, we first perform part segmentation using our compositional model for large parts, such as head, neck and torso, etc. Given these segmentation results, we can narrow down the search region for legs since legs are almost always underneath the torso. Then we segment legs by combining symmetric structure and appearance information.

Our experiment is conducted on two animal classes: horse and cow. We use a newly annotated dataset on Pascal VOC 2010 [8] which provides pixelwise semantic part annotations. We focus on segmenting fully observable animals in this paper and leave the occlusion and truncation issue for future study. Self-occlusion due to poses/viewpoints can be handled by our model. We compare our algorithm with the method that combines the state-of-the-art animal part detection [8] and object segmentation [18]. The experiment shows that our method achieves much better part segmentation than the baseline, which demonstrates the effectiveness of our method.

In summary, our contribution is threefold. Firstly, we develop a novel method for animal part segmentation by introducing a mixture of compositional models coupled with shape and appearance. Secondly, we propose an algorithm to learn the compositional models of object and part shapes given part-level pixelwise annotations. Thirdly, we develop the constrained generalized distance transform (CGDT) algorithm to achieve linear-time inference for our model.

2. Related Work

In terms of method, our work is related to [32, 31], where they used compositional model for horse segmenta-

tion. But they did not incorporate strong appearance cues into their compositional shape model, and they modeled only a few poses and viewpoints. Besides, our inference is much faster than their compositional inference algorithm. There was also work on automatically learning the compositional structure/hierarchical dictionary [33, 16], but their algorithms did not consider semantic parts and were not evaluated on challenging datasets.

In terms of task, our work is related to human parsing/clothes parsing [4, 10, 9, 29]. They generated segment proposals by superpixel/over-segmentation algorithms, and then used these segments as building blocks for whole human body by either compositional method or And-Or graph. Note that our task is inherently quite different from clothes parsing because animals often have roughly homogeneous appearance throughout the whole body while in the human parsing datasets humans often have different appearance (e.g., color) for different part due to clothes. So their superpixel/over-segmentation algorithms could not output good segment proposals for animal parts. Besides, in challenging datasets like Pascal VOC, cluttered background and unclear boundaries further degrade the superpixel quality. Therefore, the superpixel-based methods for human parsing are not appropriate for our animal part segmentation task.

Our work also bears a similarity to [35] in the spirit that a mixture of graphical models are used to capture global variation due to viewpoints/poses. But our compositional model is able to capture spatial relation between children nodes while still achieving linear complexity inference, and we develop an algorithm to learn the mixtures of compositional models. Besides, our task is part segmentation for animals of various poses and viewpoints, which appears more challenging than landmark localization for faces in [35].

There are lots of works in the literature on modeling object shape such as [25, 15, 28, 20]. But they were only aimed at object-level detection or segmentation. Furthermore, none of them combined shape representation with strong appearance information.

3. Compositional Model combining Shape and Appearance

We develop a compositional model to represent animal shape/boundary under various viewpoints and poses. We formulate the compositional part-subpart relation by a probabilistic graphical model. Let v denote the parent node which represents the part and $ch(v)$ denote the children nodes which represent the constituent subparts. The location of part v is denoted by $S_v = (x_v, y_v)$ and the locations of its subparts $ch(v)$ are denoted by $S_{ch(v)}$. The probability distribution for the part-subpart composition is modeled as a Gibbs distribution
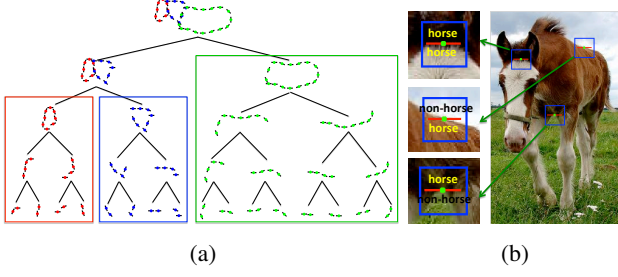
Figure 2: (a) Illustration of compositional model for a particular horse shape. Red for head, blue for neck and green for torso. Due to space limitation, the leaf nodes (oriented edgelet of eight orientations) are not shown. (b) Three types of polarity value for a leaf node with a horizontal orientation. Green dot represents center location and red line segment represents orientation. Best viewed in color.

$$P(S_{ch(v)}|S_v) = \begin{cases} \frac{1}{Z}\exp(-\psi(S_{ch(v)})), & \text{if } S_v = f(S_{ch(v)}) \\ 0, & \text{otherwise.} \end{cases}$$
(1)

Here $f(S_{ch(v)})$ is a deterministic function. In this paper, we limit the number of subparts for any part to be two, i.e., $|ch(v)| = 2$. And we set $f(S_{ch(v)}) = S_{ch(v)}/2$, which means that the location of a part is the average location of its children subparts. Potential function $\psi(S_{ch(v)})$ represents the relation between two children subparts. Let $ch(v) = (v_1, v_2)$. We have

$$\psi(S_{ch(v)}) = w_v \cdot (dx_v^2, dy_v^2),$$
(2)

where $dx_v = x_{v_2} - x_{v_1} - \Delta x_v$ and $dy_v = y_{v_2} - y_{v_1} - \Delta y_v$. Here $\Delta S_v = (\Delta x_v, \Delta y_v)$ is the location of part v's second subpart v_2 relative to its first subpart v_1. And (dx_v, dy_v) is the location displacement of second subpart v_2 relative to its anchor location.

In summary, a part node v is uniquely specified by its children $ch(v)$ and the spatial relation ΔS_v between children. In terms of the parent-children relation, our compositional model is similar to the prevailing pictorial structure [14] and deformable part model [13]. But our model is able to capture mutual relation between children.

An object can be modeled by repeating the part-subpart compositions, as shown in Figure 2 (a). Mathematically, we use a probabilistic graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ to model the object. This graph has a hierarchical structure with levels $l \in \{1, ..., L\}$, and $\mathcal{V} = \cup_{l=1}^{L} \mathcal{V}_l$, where $\mathcal{V}_l$ denotes the node set at level-l. At the top level (level-L), there is one root node, representing the object (i.e., $|\mathcal{V}_L| = 1$). The leaf node is $v \in \mathcal{V}_1$. If a part node v is at level-l, i.e., $v \in \mathcal{V}_l$, then its children subparts must be at level-$(l-1)$, i.e., $ch(v) \subset \mathcal{V}_{l-1}$. And as mentioned above, for any part,

we limit the number of subparts to be two. So there are in total 2^{L-1} leaf nodes and $2^L - 1$ nodes in the graph $\mathcal{V}$. Any part-subpart pair constructs an edge in this graph, i.e., $(v, t) \in \mathcal{E}$ if $t \in ch(v)$. There is no edge between two children subparts of one parent part, i.e., $(s, t) \notin \mathcal{E}$ if $s \in ch(v), t \in ch(v)$. Thus the hierarchical graph $\mathcal{G}$ has a tree-like structure[1]. The probability distribution for the object is specified by products of part-subpart probabilities

$$P(S_{\mathcal{V}}) = \prod_{v \in \mathcal{V} \setminus \mathcal{V}_1} P(S_{ch(v)}|S_v)P(S_{\mathcal{V}_L}).$$
(3)

We assume $P(S_{\mathcal{V}_L})$ is uniformly distributed. The compositional model introduced above can be viewed as a prior shape model for the object since it characterizes the spatial relation between parts and subparts. To specify a fully generative model for the object, we need to define a likelihood function

$$P(\mathbf{I}|S_{\mathcal{V}}) = \frac{1}{Z}\exp(-\sum_{v \in \mathcal{V}} \phi(S_v, \mathbf{I}))).$$
(4)

The MAP inference performs

$$\max_{S_{\mathcal{V}}} P(S_{\mathcal{V}}|\mathbf{I}) \propto P(\mathbf{I}|S_{\mathcal{V}})P(S_{\mathcal{V}}),$$
(5)

which is equivalent to minimizing the energy

$$E(\mathbf{I}) = \min_{S_{\mathcal{V}}} E(S_{\mathcal{V}}, \mathbf{I}) = \sum_{v \in \mathcal{V}} \phi(S_v, \mathbf{I}) + \sum_{\substack{v \in \mathcal{V} \setminus \mathcal{V}_1 \\ S_v = f(S_{ch(v)})}} \psi(S_{ch(v)}).$$
(6)

3.1. Feature for Unary Term

Next we explain the potential function (unary term) which interacts with the image. We assume that the parameters are shared by parts which will be discussed in detail in Section 5.2. Specifically, the potential function for leaf node $v \in \mathcal{V}_1$ is modeled as

$$\phi(S_v, \mathbf{I}) = \phi^{\text{edge}}(S_v, \mathbf{I}) + \phi^{\text{app}}(S_v, \mathbf{I}).$$
(7)

The first term $\phi^{\text{edge}}(S_v, \mathbf{I})$ characterizes how well the orientation at location S_v in the image matches the model orientation θ_v. In the experiment we use the gPb edge detection result [2] which outputs pixelwise confidence score for eight orientations. Thus

$$\phi^{\text{edge}}(S_v, \mathbf{I}) = w_v^{\text{edge}} \cdot gPb(\theta_v, S_v, \mathbf{I}).$$
(8)

To incorporate appearance information, each leaf node v is associated with an polarity value a_v (specified by the model as θ_v) indicating which side of the leaf node is object side,

---

[1]Precisely, it is not a tree since two children are connected. But we prefer calling it tree structure in this paper for explanation purposes.

and which side is non-object (background) side. We extract a square centered at location S_v, and obtain the object-side region and non-object-side region based on the orientation, as shown in Figure 2 (b). We use the semantic labeling result [24] as the appearance feature. It gives pixelwise segmentation result for 34 classes including 20 object classes from Pascal VOC and another 14 background classes. Each pixel is associated with a 34-dimensional vector with each component being the confidence score for the corresponding class. We average the feature vector of object-side region and non-object-side region, and then concatenate them to make a 68-dimensional feature vector denoted by $SemLab(\theta_v, S_v, \mathbf{I})$. We use the confidence scores of all classes to deal with inaccurate semantic labeling and context information. Thus we have

$$\phi^{\mathrm{app}}(S_v, \mathbf{I}) = w_v^{\mathrm{app}} \cdot SemLab(\theta_v, S_v, \mathbf{I}). \qquad (9)$$

For the non-leaf node $v \in \mathcal{V}_l, l > 1$, the unary term $\phi(S_v, \mathbf{I})$ indicates the confidence of part v being at location S_v. The confidence score can be from some part detection algorithm [8] for animals.

$$\phi(S_v, \mathbf{I}) = w_v^{\mathrm{part}} \cdot PartDet(S_v, \mathbf{I}). \qquad (10)$$

For example, if v represents the horse head, $PartDet(S_v, \mathbf{I})$ can be the horse head detection score.

3.2. Mixture of Poses and Viewpoints

We have so far introduced a compositional model for animal of one single viewpoint and pose. In order to model various poses and viewpoints, we use a set of nodes at the top level ($v \in \mathcal{V}_L$), each of which represents animal shape from one viewpoint and pose. Basically we use a mixture model with each mixture being a node at the top level. Section 5.1 will introduce how to learn the mixtures.

4. Inference for Compositional Model

Given an image, the goal of inference is to find the best mixture $v \in \mathcal{V}_L$ (i.e. the best viewpoint and pose) and specify locations of all its descendants $S_{tree(v)}$, especially locations of all leaf nodes as boundary landmarks. Then we can connect adjacent landmarks of each semantic part to give part segmentation result. Basically, for each mixture $v \in \mathcal{V}_L$, we solve the minimization problem (6) by standard dynamic programming on the tree $tree(v)$. And then we select the mixture with the minimal energy as the best mixture. The dynamic programming algorithm involves a bottom-up process starting from the leaf nodes to find the minimal energy, which is followed by a top-down process to find the best configuration.

The search is done over every pixel in the image grid. Denote the image grid by $\mathcal{D} = \{1, ..., W\} \times \{1, ..., H\}$, and the size of the image grid is $|\mathcal{D}| = W \times H$. The core of

---

Algorithm 1 The CGDT algorithm

Initialization:
    $range(1) = u^{-1}(1);\ range(2) = l^{-1}(1);$
    $idx(1) = 1;\ k = 1;$
Process:
1: For $z = 2$ to n
2:     $s = \frac{(g(z) + h^2(z)) - (g(idx(k)) + h^2(idx(k)))}{2h(z) - 2h(idx(k))};$
3:     Project s onto interval $[u^{-1}(z), l^{-1}(z)]$;
4:     While $s \le range(k)$
5:         k = k-1;
6:         $s = \frac{(g(z) + h^2(z)) - (g(idx(k)) + h^2(idx(k)))}{2h(z) - 2h(idx(k))};$
7:         Project s onto interval $[u^{-1}(z), l^{-1}(z)]$;
8:     end
9:     If $s > range(k + 1)$
10:         $k = k + 1;\ idx(k) = z;$
11:         $range(k + 1) = l^{-1}(z);$
12:     Else
13:         $k = k + 1;\ idx(k) = z;$
14:         $range(k) = s;\ range(k + 1) = l^{-1}(z);$
15:     end
16: end
17: Fill in the value of $\gamma(x)$ using $range(k)$ and $idx(k)$.

---

dynamic programming is to solve the following minimization problem for each non-leaf node

$$
\begin{aligned}
E(S) &= \min_{\substack{\{S_1, S_2\} \\ 2S = S_1 + S_2}} \phi(S_1, S_2) + E_1(S_1) + E_2(S_2) \\
&= \min_{\substack{\{S_1\} \\ 2S - S_1 \in \mathcal{D}}} \phi(S_1, 2S - S_1) + E_1(S_1) + E_2(2S - S_1).
\end{aligned}
$$
$$(11)$$

Here S, S_1 and S_2 denote the locations of the parent (part) node and the two children (subpart) nodes respectively. $E(S)$, $E(S_1)$ and $E(S_2)$ denote the energy functions in the dynamic programming. For simplicity, we drop the subscript v since it applies to every non-leaf node. Exact solution of problem (11) requires quadratic complexity $O(|\mathcal{D}|^2)$, which is too slow in practice. This drives us to design an algorithm to achieve linear complexity $O(|\mathcal{D}|)$. Therefore we approximate problem (11) by

$$E(S) \approx \min_{\substack{\{S_1\} \\ 2S - S_1 \in \mathcal{D}}} \phi(S_1, 2S - S_1) + E_1(S_1) + E_2(2S - S_1^*),$$

$$S_1^* = \arg \min_{\substack{\{S_1\} \\ 2S - S_1 \in \mathcal{D}}} \phi(S_1, 2S - S_1) + E_1(S_1). \qquad (12)$$

The reason of making such approximation is that we can then solve problem (12) efficiently in linear time using the constrained generalized distance transform algorithm developed in Section 4.1. We will validate this approximation by experiment in Section 7.1.

$$(x_1^*, y_1^*) = \arg \min_{\substack{\{y_1\} \\ 1 \le 2y - y_1 \le H}} \left\{ 4w^y(y - y_1 - \frac{\Delta y}{2})^2 + \min_{\substack{\{x_1\} \\ 1 \le 2x - x_1 \le W}} 4w^x(x - x_1 - \frac{\Delta x}{2})^2 + E_1(x_1, y_1) \right\}. \quad (13)$$

4.1. Constrained Generalized Distance Transform (CGDT) Algorithm

First note that since the variables $S_1 = (x_1, y_1)$ are separable, we can translate the 2-dimensional problem (12) into two 1-dimensional problems by first minimizing one variable (x_1) and then minimizing the other one (y_1), as shown in Equation (13). Next we show how to efficiently solve these two similar 1-dimensional subproblems. To this end, we consider a slightly more general problem of the form

$$\gamma(x) = \min_{l(x) \le z \le u(x)} (x - h(z))^2 + g(z), \quad (14)$$

where $h(z)$, $u(x)$ and $l(x)$ are all non-decreasing. In Equation (13), for the inner minimization, we set $h(z) = z + \frac{\Delta x}{2}$ and $l(x) = 2x - W, u(x) = 2x - 1$; and for the outer minimization, we set $h(z) = z + \frac{\Delta y}{2}$ and $l(y) = 2y - H, u(y) = 2y - 1$. Note that problem (14) becomes the ordinary generalized distance transform [12] if we ignore the constraint $l(x) \le z \le u(x)$. Inspired by [12], $\gamma(x)$ can be viewed as the lower envelope of a set of truncated parabolas $(x - h(z))^2 + g(z)$ with the truncation being $u^{-1}(z) \le x \le l^{-1}(z)$. The algorithm performs in two steps. In the first step we obtain the lower envelope of all the truncated parabolas by computing the boundary points between adjacent selected parabolas while keeping the truncation constraint being satisfied. In the second step we fill in the value $\gamma(x)$ using the obtained lower envelope from step one. Algorithm pseudocode is provided in Algorithm 1, where we use $range(k)$ and $range(k+1)$ to indicate the range of k-th parabola in the lower envelope, and $idx(k)$ to indicate the grid location z of the k-th parabola in the lower envelope.

5. Learning for Compositional Model

5.1. Structure Learning

Structure learning refers to learning the hierarchical graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ to represent the animal and part shapes under various poses and viewpoints. Specifically, for each non-leaf node v, we need to learn the part-subpart relation $ch(v)$ and ΔS_v; and for each leaf nodes $v \in \mathcal{V}_1$, we need to learn the orientation θ_v and polarity a_v. We consider eight orientations which are equally distributed from 0 to π, and three polarity values for each orientation which represent object region on one side, object region on the other side, and object region on both sides respectively, as shown in Figure 2 (b). Thus there are in total 24 types of leaf nodes

at level one. Note that leaf nodes are shared across different mixtures.

We use compositional models to represent big semantic parts such as head, neck and torso, and we discuss segmenting legs in Section 6. The structure learning algorithm proceeds in the following four steps.

1. Clustering: Given part-level annotations, we extract the masks for head, neck and torso and assign them different values (1 for head, 2 for neck, and 3 for torso). Then we resize each example by the maximal side length. We apply the K-medoids clustering algorithm to find K representative shapes from the training data. And we will build K compositional mixtures based on the K representative shapes.

2. Sampling: We evenly sample fixed number of landmarks along the boundary of each semantic part.

3. Matching: We match each landmark to one of the 24 leaf nodes.

4. Composing: Starting from the landmarks (leaf nodes), we compose each two adjacent nodes (children) into a higher-level node (parent) and record the spatial relation between the two children nodes. The parent location is the average of two children locations. We run this procedure level-by-level up to the top level.

5.2. Parameter Learning

The parameters of the compositional model are w_v and w_v^{part} for non-leaf nodes, and w_v^{edge} and w_v^{app} for leaf nodes. To reduce the model complexity, we assume that parameters are shared by parts. So the parameter vector becomes $\mathbf{w} = (w, w^{\text{part}}, w^{\text{edge}}, w^{\text{app}})$. These parameters strike a balance between the prior shape (w), appearance cues (w^{app}), orientation confidence (w^{edge}) and part confidence (w^{part}). The sharing allows us to learn the model parameters using a small number of training data. Note that the energy function $E(S_{\mathcal{V}}, \mathbf{I}; \mathbf{w})$ is of the form

$$E(S_{\mathcal{V}}, \mathbf{I}; \mathbf{w}) = \mathbf{w} \cdot \phi(S_{\mathcal{V}}, \mathbf{I}). \quad (15)$$

The training dataset is denoted by $\{(\mathbf{I}_i, y_i)\}_{i=1}^n$, where $y_i \in \{+1, -1\}$. The positive examples refer to object bounding box images and negative examples refer to bounding box images of other objects. Since we do not have the location information for all parts/subparts $S_{\mathcal{V}}$, we adopt latent SVM for learning parameters $\mathbf{w}$.

$$\min_{\mathbf{w}} \quad \frac{1}{2}||\mathbf{w}||^2 + C \sum_{i=1}^n \max(0, 1 - y_i F(\mathbf{I}_i; \mathbf{w})), \quad (16)$$

where the score function is defined as $F(\mathbf{I}_i; \mathbf{w}) = -\min_{S_\mathcal{V}} E(S_\mathcal{V}, \mathbf{I}_i; \mathbf{w})$.

6. Segmenting Legs

Considering the extremely high variability of animal legs, we take a coarse-to-fine approach to segment legs. Specifically, after segmenting the animal body (head, neck, torso), we can narrow down the search region for legs since we know that most of the time the legs appear underneath the torso. Then in the refined search region, we detect symmetric regions using algorithm in [21] since animal legs often have roughly symmetric structure. Next we compute a confidence score for each detected symmetric region R, and make prediction by thresholding this score.

$$\text{score}(R) = w^{\text{obj}} \cdot fea(R). \quad (17)$$

Here w^{obj} is the parameters corresponding to the features extracted within object region, i.e., the first half of w^{app}. And $fea(R)$ is the average 34-dimensional feature vector within region R.

7. Experiments

In this section, we will report part segmentation results for horse and cow. We also conduct some diagnostic experiments for our model. In addition, we validate by experiment that our approximate inference is much faster than exact inference while losing little accuracy.

Dataset: We use a newly annotated dataset on Pascal VOC 2010 [8] to evaluate our part segmentation algorithm. It provides pixelwise semantic part annotations for each object instance. Since we focus on non-occlusion and non-truncation case, for each animal class we manually select the fully observable animal instances in both trainval and test set. We use this refined dataset for training and testing, and we will release it. For horse and cow, there are roughly 150 fully observable bounding box images in trainval and test respectively. Considering the various poses and viewpoints of animals and cluttered background in Pascal VOC images, we believe the fully observable animal bounding box images are a suitable testbed for our algorithm.

We use the bounding box images with part annotations in the Pascal trainval set for structure learning. As for parameter learning, we use the bounding box images from Pascal VOC trainval set as positive examples and randomly select a subset of bounding box images of other object classes from Pascal VOC trainval set as negative examples. We use the bounding box images from Pascal VOC test set for testing.

Setup: We consider head, neck, torso and leg as semantic parts. In the structure learning, we set the number of boundary landmarks to be 8 for head, 8 for neck and 16 for torso. Thus each compositional tree has 6 levels and 32 leaf nodes. The head node and neck node are at the 4-th level
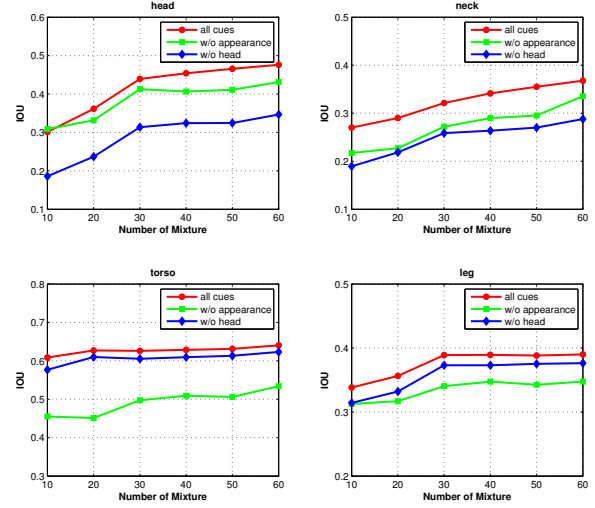

Figure 3: The performance variation with the number of mixtures for four semantic parts. The effect of having appearance cues and head cues are also shown in this figure.

and torso node is at the 5-th level. We only consider the head part score (i.e., w_v^{part} is non-zero only if v refers to the head part) since head is the most discriminative part for animals. Our algorithm outputs the best mixture and locations of all parts/subparts. We only use the locations of all leaf nodes as boundary landmarks. We connect the adjacent leaf nodes of each semantic part to make a closed contour as part segmentation result. We use intersection-over-union (IOU) as the performance measure.

7.1. Efficient Inference

Recall that we make approximations (12) in order to allow efficient linear-time inference. We now provide results demonstrating that we lose little in accuracy and gain much in speed by approximation. Let $E(\mathbf{I})$ denote the exact minimal energy (quadratic complexity) and $\tilde{E}(\mathbf{I})$ denote the minimal energy by our efficient approximate algorithm (linear complexity). We measure the error by $\frac{(E(\mathbf{I}) - \tilde{E}(\mathbf{I}))}{E(\mathbf{I})}$. We compute this error on all test images of horses and get 0.53% average error. Furthermore, we compute the average location difference of all leaf nodes between exact inference and our approximate inference algorithm. We get on average 1.78 pixels error and 1.11% if normalized by maximal side length. The above results show that our approximation is extremely accurate. As for the speed, the average parsing time per image (resize maximal side length to 160) is about 10 seconds using our inference algorithm while the average parsing time for exact inference is about 10 minutes per image. This demonstrates the significant speedup by our fast approximate inference algorithm.

7.2. Model Diagnostics

Our diagnostic experiment is based on horse images.

Number of Mixtures: The structure learning algorithm uses K-medoids clustering to find K representative shapes. Figure 3 shows how the segmentation performance varies with respect to the parameter K for each semantic part. Intuitively, as the number of mixture increases, our mixtures of compositional models are able to capture more deformations and variations of animal and part boundaries. Therefore, the segmentation performance improves with the number of mixtures. Particularly, small parts (head and neck) benefit significantly from increasing mixture number.

Appearance and Head cues: We can also see from Figure 3 that the performance drops if we do not use appearance cues from semantic labeling or head cues from animal part detection. This result indicates that combining appearance and part information are necessary for localizing boundaries although they are not always correct.

Deformation Ability of Compositional Model: Figure 4 shows that each mixture deals with local deformation, and different mixtures handle large global variation due to poses and viewpoints. Thus our mixtures of compositional models are able to capture various shape variations.

Failure Cases: Figure 6 shows three typical failure cases. The reason for (a) is because the horse is in very rare pose which cannot be captured by any mixture. The reason for (b) is because the semantic labeling result is wrong and the horse boundary is unclear due to dark lighting. Incorrect body segment often leads to wrong leg segmentation (e.g., case (a) and (b)). In (c), the legs are mistakenly segmented although the horse body segment is correct. This is because both the detected symmetric structure (red region on the image) and the semantic labeling result are not correct.

7.3. Comparison

Baseline: There has been lack of work on semantic part segmentation for animals. But there are part-based object detection work [8] that is able to output part-level bounding boxes. There is also many object segmentation works that give object-level segments. Therefore, it is straightforward to combine part detection and object segmentation to output part-level segmentation result. We use the state-of-the-art object segmentation algorithm [18] in the experiment. Take the head as an example. We treat certain part of the object segment that lies inside the head bounding box as the head segment. This method is our first baseline. Furthermore, to better model the part shape, we learn a mixture of 10 masks for each part and compute the intersection of the best part mask and the object segment. To select the best mask for each part, we implemented an oracle method which selects the part mask that gives the best IOU with the ground-truth part segmentation. This is our second baseline.

We conduct our experiments on two animal classes:

Method	head	neck+torso	leg
Our model	41.55	60.98	30.98
PD+OS	26.77	53.79	11.18
Mask+OS	33.19	56.69	11.31
PD+GT	38.66	60.63	19.36

Method	head	neck	torso	neck+torso	leg
Our model	47.21	38.01	61.02	66.74	38.18
PD+OS	37.32	N/A	N/A	60.35	27.47
Mask+OS	41.84	N/A	N/A	63.31	21.38
PD+GT	56.64	N/A	N/A	67.96	40.95

Table 1: Part segmentation result for horses (bottom) and cows (top). The performance measure is IOU (%). PD+OS refers to the method that combines part detection bounding box and object segmentation (first baseline). Mask+OS refers to the method that uses oracle mask selection and object segmentation (second baseline). PD+GT refers to the method that combines part detection bounding box and groundtruth segmentation.

horse and cow. Table 1 shows quantitative results and Figure 5 gives some part segmentation visualizations. The horse model has 60 mixtures and cow model has 40 mixtures. Since the part detection method [8] treats neck+torso as one part, we do not have detection bounding box for neck and torso separately. For cows we did not split neck and torso since the cow neck is always small, which is in contrast to the long horse neck. We can see that our part segmentation results are significantly better than the two baseline methods (PD+OS and Mask+OS). We can also see that our results are only a little lower than the method (PD+GT) that combines the part detection bounding box and the groundtruth animal segmentation. This further validates the effectiveness of our method.

8. Conclusion

In this paper, we built a mixture of compositional models combining shape and appearance for animal part segmentation task. We proposed a novel structure learning algorithm to learn the mixture of compositional trees which are able to represent animal shapes of various poses and viewpoints. We also developed a linear complexity algorithm to significantly speed up the inference of the compositional model. We tested our method for horse and cow on the Pascal VOC dataset. The experimental results showed that our method achieves much better part segmentation results than the baseline methods. As for the future work, we will deal with occlusion and truncation issue, and enable part sharing when learning the compositional models.
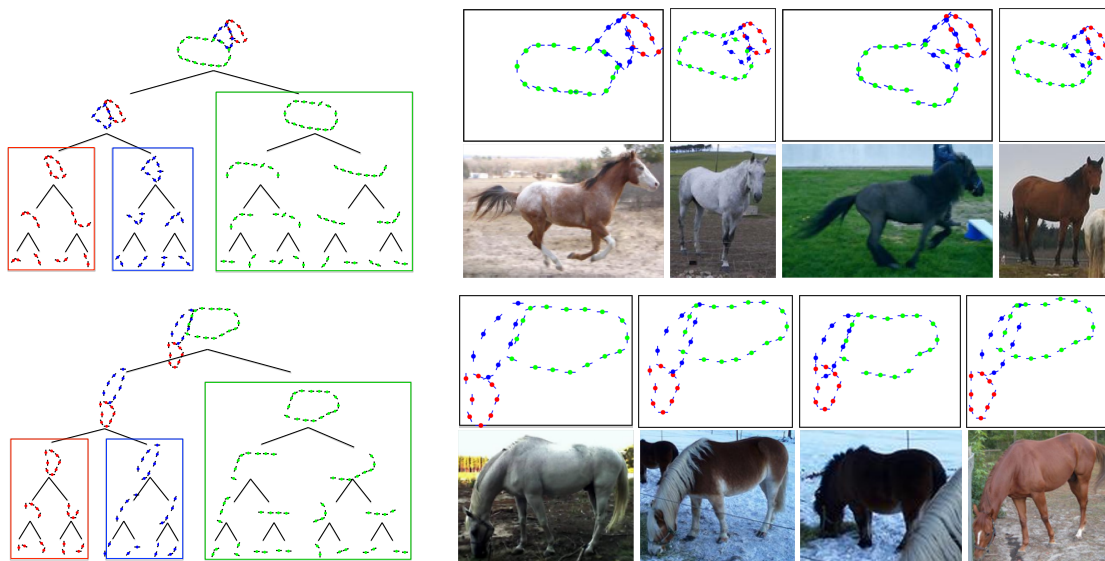
Figure 4: Two mixtures and corresponding landmark localization results. For each mixture, the left figure is the compositional model, the top row on the right is the landmark localization results, and the bottom row on the right is the input images. We can see that each mixture deals with local deformation, and different mixtures handle large variation due to poses and viewpoints.
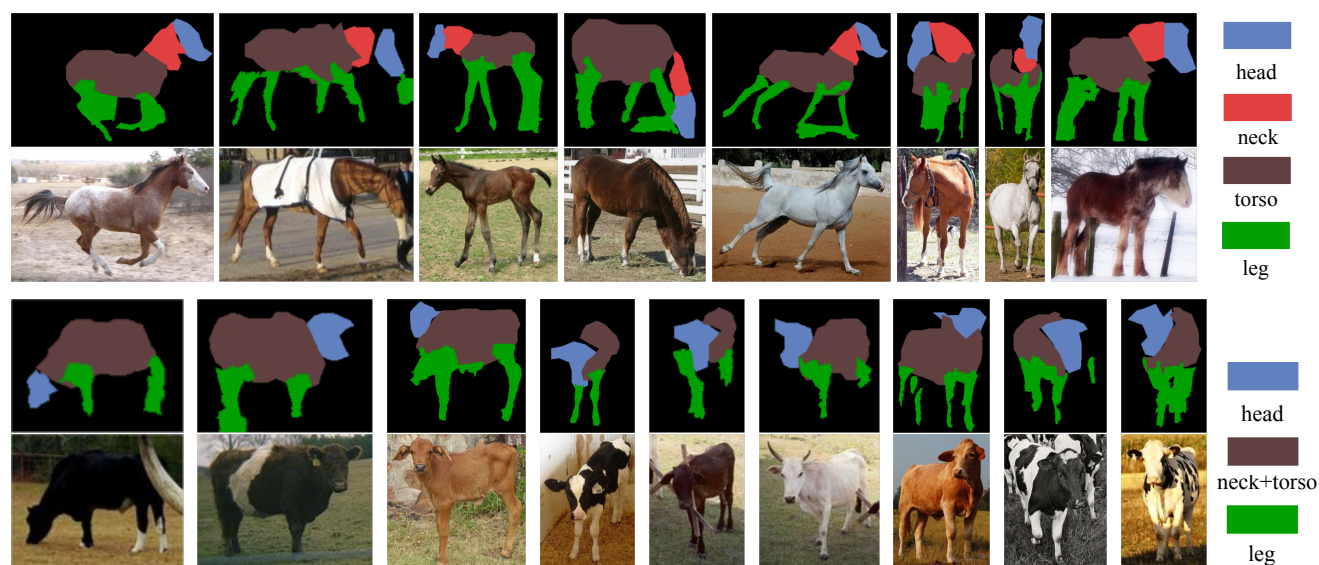

Figure 5: Typical semantic part segmentation results from various viewpoints and poses for horses (top) and cows (bottom). Best viewed in color.
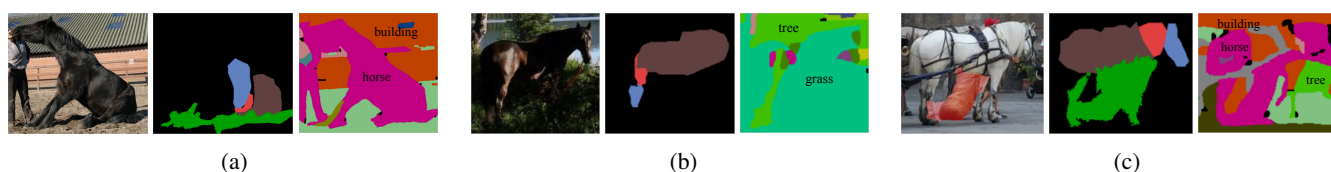

(a)   (b)   (c)

Figure 6: Three typical failure cases. For each case, left is image, middle is part segmentation result, and right the semantic labeling result. (a) rare pose. (b) mistaken semantic labeling and unclear boundary. (c) correct body segment but wrong leg segment due to mistaken semantic labeling and symmetric structure (red region on the image).

References

[1] P. Arbeláez, B. Hariharan, C. Gu, S. Gupta, L. Bourdev, and J. Malik. Semantic segmentation using regions and parts. In *CVPR*, 2012. 1

[2] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *PAMI*, 33(5):898–916, 2011. 1, 3

[3] P. N. Belhumeur, D. W. Jacobs, D. Kriegman, and N. Kumar. Localizing parts of faces using a consensus of exemplars. In *CVPR*, 2011. 1

[4] Y. Bo and C. C. Fowlkes. Shape-based pedestrian parsing. In *CVPR*, 2011. 1, 2

[5] J. Carreira, R. Caseiro, J. Batista, and C. Sminchisescu. Semantic segmentation with second-order pooling. In *ECCV*, 2012. 1

[6] J. Carreira and C. Sminchisescu. Cpmc: Automatic object segmentation using constrained parametric min-cuts. *PAMI*, 34(7):1312–1328, 2012. 1

[7] L.-C. Chen, G. Papandreou, and A. L. Yuille. Learning a dictionary of shape epitomes with applications to image labeling. In *ICCV*, 2013. 1

[8] X. Chen, R. Mottaghi, X. Liu, S. Fidler, R. Urtasun, and A. Yuille. Detect what you can: Detecting and representing objects using holistic models and body parts. In *CVPR*, 2014. 2, 4, 6, 7

[9] J. Dong, Q. Chen, X. Shen, J. Yang, and S. Yan. Towards unified human parsing and pose estimation. In *CVPR*, 2014. 1, 2

[10] J. Dong, Q. Chen, W. Xia, Z. Huang, and S. Yan. A deformable mixture parsing model with parselets. In *ICCV*, 2013. 1, 2

[11] S. A. Eslami and C. Williams. A generative model for parts-based object segmentation. In *NIPS*, 2012. 1

[12] P. Felzenszwalb and D. Huttenlocher. Distance transforms of sampled functions. Technical report, Cornell University, 2004. 2, 5

[13] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *PAMI*, 32(9):1627–1645, 2010. 1, 3

[14] P. F. Felzenszwalb and D. P. Huttenlocher. Pictorial structures for object recognition. *IJCV*, 61(1):55–79, 2005. 3

[15] V. Ferrari, F. Jurie, and C. Schmid. From images to shape models for object detection. *IJCV*, 87(3):284–303, 2010. 2

[16] S. Fidler and A. Leonardis. Towards scalable representations of object categories: Learning a hierarchy of parts. In *CVPR*, 2007. 2

[17] S. Fidler, R. Mottaghi, A. L. Yuille, and R. Urtasun. Bottom-up segmentation for top-down detection. In *CVPR*, 2013. 1

[18] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Simultaneous detection and segmentation. In *ECCV*, 2014. 2, 7

[19] Y. Jin and S. Geman. Context and hierarchy in a probabilistic image model. In *CVPR*, 2006. 1

[20] I. Kokkinos and A. Yuille. Inference and learning with hierarchical shape models. *IJCV*, 93(2):201–225, 2011. 1, 2

[21] T. S. H. Lee, S. Fidler, and S. Dickinson. Detecting curved symmetric parts using a deformable disc model. In *ICCV*, 2013. 6

[22] J. Liu and P. N. Belhumeur. Bird part localization using exemplar-based models with enforced pose and subcategory consistency. In *ICCV*, 2013. 1

[23] W. Lu, X. Lian, and A. Yuille. Parsing semantic parts of cars using graphical models and segment appearance consistency. In *BMVC*, 2014. 1

[24] R. Mottaghi, X. Chen, X. Liu, N.-G. Cho, S.-W. Lee, S. Fidler, R. Urtasun, and A. Yuille. The role of context for object detection and semantic segmentation in the wild. In *CVPR*, 2014. 4

[25] J. Shotton, A. Blake, and R. Cipolla. Contour-based learning for object detection. In *ICCV*, 2005. 2

[26] A. Thomas, V. Ferrari, B. Leibe, T. Tuytelaars, and L. Van Gool. Using recognition to guide a robot's attention. *Robotics: Science and Systems*, 2008. 1

[27] A. Toshev and C. Szegedy. Deeppose: Human pose estimation via deep neural networks. In *CVPR*, 2014. 1

[28] Y. N. Wu, Z. Si, H. Gong, and S.-C. Zhu. Learning active basis model for object detection and recognition. *IJCV*, 90(2):198–235, 2010. 2

[29] K. Yamaguchi, M. H. Kiapour, L. E. Ortiz, and T. L. Berg. Parsing clothing in fashion photographs. In *CVPR*, 2012. 1, 2

[30] Y. Yang and D. Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *CVPR*, 2011. 1

[31] L. Zhu, Y. Chen, C. Lin, and A. Yuille. Max margin learning of hierarchical configural deformable templates (hcdts) for efficient object parsing and pose estimation. *IJCV*, 93(1):1–21, 2011. 2

[32] L. Zhu, Y. Chen, and A. Yuille. Learning a hierarchical deformable template for rapid deformable object parsing. *PAMI*, 32(6):1029–1043, 2010. 1, 2

[33] L. Zhu, C. Lin, H. Huang, Y. Chen, and A. Yuille. Unsupervised structure learning: Hierarchical recursive composition, suspicious coincidence and competitive exclusion. In *ECCV*. 2008. 2

[34] S.-C. Zhu and D. Mumford. *A stochastic grammar of images*. Now Publishers Inc, 2007. 1

[35] X. Zhu and D. Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *CVPR*, 2012. 2