Beyond Holistic Object Recognition:
Enriching Image Understanding with Part States

Cewu Lu[1], Hao Su[2], Yonglu Li[1], Yongyi Lu[3], Li Yi[4], Chi-Keung Tang[3], Leonidas J. Guibas[4]

[1]Shanghai Jiao Tong University
[2]University of California San Diego
[3]Hong Kong University of Science and Technology
[4]Stanford University

{lucewu, yonglu_li}@sjtu.edu.cn haosu@eng.ucsd.edu {yluaw, cktang}@cse.ust.hk
{ericyi, guibas}@cs.stanford.edu

Abstract

Important high-level vision tasks require rich semantic descriptions of objects at part level. Based upon previous work on part localization, in this paper, we address the problem of inferring rich semantics imparted by an object part in still images. Specifically, we propose to tokenize the semantic space as a discrete set of part states. Our modeling of part state is spatially localized, therefore, we formulate the part state inference problem as a pixel-wise annotation problem. An iterative part-state inference neural network that is efficient in time and accurate in performance is specifically designed for this task. Extensive experiments demonstrate that the proposed method can effectively predict the semantic states of parts and simultaneously improve part segmentation, thus benefiting a number of visual understanding applications. The other contribution of this paper is our part state dataset which contains rich part-level semantic annotations.

1. Introduction

Recently there has been growing interest in understanding the detailed semantics of images, because important high-level vision tasks such as human-object interaction, robotic manipulation and image captioning require object understanding beyond holistic object recognition. In particular, rich description of objects at part level is necessary since interaction among objects are often manifested as contact of the pertinent object parts.

Existing work has almost exclusively focused on part localization [26]. However, localization of object parts only scratched the surface of understanding the rich informa-tion embodied in encapsulated by the object parts. In fact, through scrutinizing the visual appearance of object parts, rich semantic information about a single object and the relationships among multiple objects can be revealed. For instance, by seeing observing a person's *hand* turning the *door knob*, we infer that this person may attempt to open a door and enter a room. Figure 1 gives more examples of part semantics, ranging from including functionality, geometry relationship, affordance, moment situation (a.k.a fluent as in [12]), to interaction. We also notice that high-level semantics on parts are important for robotic manipulation tasks. For example, affordance and interaction modeling are explored in [20] and [21] respectively.

This paper makes a significant attempt to advance the next step: we propose to tokenize the semantic space of relevant object parts into a discrete set of *part state*. Specifically, a part of an object is associated with a set of states, each of which is characterized by a phrase that describes its semantic meaning.

To implement this idea, we first have to define a vocabulary of part states. We address this issue by resorting to natural language processing. We collect phrase-level human descriptions on the relevant object parts in scene images, specifically, PASCAL VOC 2010 images. Though simple, these phrase-level descriptions carry rich semantics (see Figure 1). The description of a particular part is readily categorized into different discrete groups, where each group belongs to a part-state with a summarized phrase. Note that our part state annotations are object-centric, i.e., only the object containing the part of interest is described by its category name, all the other objects are referred to as "something else". Figure 2(a) shows an example of part state generation, and (b) illustrates an example of part states.

There are two desirable features associated with our part states implementation: semantic tokenization and the

| (a) functionality | (b) geometry relationship | (c) affordance | (d) moment situation | (e) interaction |

Figure 1: Parts carry rich semantics: functionality, geometry relationship, affordance, moment situation, interaction. Out-of-the-scope objects are referred to as "sth", short form for "something else".
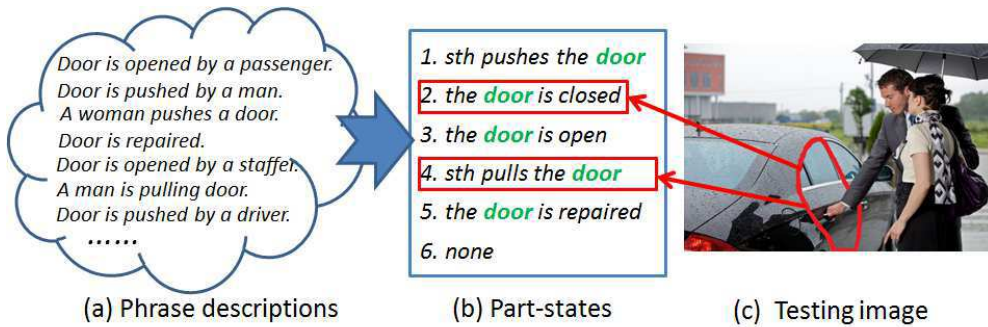

Figure 2: The pipeline of part state generation for the part door (in the car category). (a) Phrase descriptions from different annotators for different images, (b) manually summarizing of descriptions into different part states with phrases, (c) a test image falls onto two part states with indices 2 and 4.

"something else" trick. Due to the small semantic variation of simple parts, we can measure the semantic space on parts with rich descriptions, and categorize them into a manageable set of semantic tokens, or part states. This leads to a conventional multi-class labeling problem that can be objectively evaluated. Therefore, our work is different from some more complex tasks like image captioning [25], where each person has different and possibly subjective descriptions on the same visual data. The other feature, which is termed as the "something else" trick, allows attention to be paid only on target objects which greatly simplifies the complex problem. For non-target objects we refer to them as "something else". This trick resonates with how infants learn to grasp an object unseen before [10]. That is, to learn the main concept "hand grasps something", we do not need to learn exactly what that something is. This trick will be applied in learning our part states, which can avoid a huge number of semantically redundant part states (e.g. "hand grasps apple", "hand grasps orange", "hand grasps lemon", etc).

Computationally, our goal is to predict part states and simultaneously correct part localization errors given an object image. The inherent challenge is that while part state apparently depends on local part information, it is also related to

the holistic object appearance. Therefore, we propose to use an RGB-S image which concatenates the input RGB image and its part-segmented image (S). The RGB image provides holistic object information while the part-segmented image provides local part information. With the input RGB-S image, we propose an iterative part state inference network which iteratively optimizes the part-segmented image under the guidance of the part state prior by minimizing the part state prediction error. Part segment shapes and part states are closely related to each other and thus a better part segmentation will lead to less part state prediction error.

To benchmark our performance, we construct a dataset with pixel-wise part labels and part states, which will be published alongside with this paper. Extensive experiments show that our proposed iterative part state inference network produces excellent part state results.

2. Related Work

Holistic object recognition. Conventional object recognition aims at object category labeling given a test image. Earlier work such as visual word coding [27] uses statistical information of local patches. The deformable part model, now known as DPM [6] uses part relationship and part appearance. Deep learning has recently made signif-
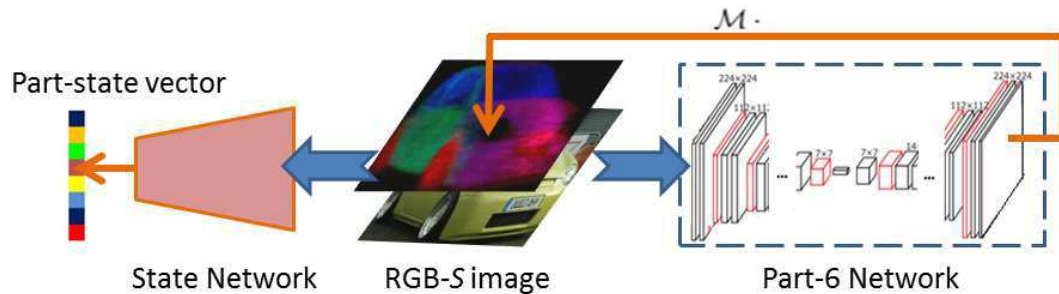
Figure 3: Iterative Part-state Inference Network (ISIN) architecture. The RGB-S image is the input to the part-6 network and the state network. The current S image (output of the part-6 network) stacked with the input RGB image is the input RGB-S image to the next iteration.

icant contributions to object recognition. Representative network architectures include AlexNet [11], VGG [22] and ResNet [8]. Excellent object detection methods founded on one of these architectures include RCNN [7] and Faster RCNN [18]. Object segmentation can be considered as a special holistic object recognition task [13, 17] which cuts the object from the background.

Layout of object parts. Object parts layout has been used to provide sub-object level information. The specific problem closely related to object parts detection is human pose estimation [24, 23, 2] where different human parts (e.g., head, body, hands, legs) need to be localized. In [4], a separate representation was respectively proposed for holistic object and body parts, and a fully connected model was used to optimize their arrangement. The model was applied to the six animal categories and achieved a better object representation performance. In [26], to segment object parts, a mixture of compositional models was used to represent the object boundary and the boundaries of the semantic parts. This compositional model incorporates edge, appearance, and semantic parts. The above methods localize parts only, but not in-depth explore semantics on them.

Image Captioning. Our part state can be considered as a "caption" on the associated part region. Here, we survey a number of works on image captioning. In [25], a generative model was presented that is based on a deep recurrent architecture. Combining the recent advanced machine translation techniques, the model was trained to maximize the likelihood of the target description sentence on the training images. In [9], inter-modal correspondences were proposed between language and visual data. To some degree image captioning explores high-level image semantics. However, image captions vary from person to person and are difficult to be objectively measured.

3. Object Part-state Dataset

None of the existing datasets provides the description of part states, therefore, we build a dataset with part state de-

scriptions for training and benchmarking our learning-based system. Our part state dataset is built on top of the part localization dataset from UCLA [26], which provides pixel-wise part membership annotation on the PASCAL VOC 2010 dataset. We refer to our part state dataset as *PASCAL VOC 2010 Part State Dataset*.

Our dataset covers 15 object categories, 104,965 parts and 856 part states in total, annotated from 19,437 object images. Some parts, such as eyes and ears, are too small to detect individually, so we merge them together into one bigger part with a detectable size, e.g., eyes and ears are merged to be parts of heads. We have also fixed missing and wrong annotations. We follow [26] for the training and testing splits. We asked 15 subjects to annotate the UCLA part dataset [26] with phrase descriptions without any given constraints. Then, we manually categorize these raw descriptions into different groups where each group is indexed by a part state (with a phrase description) according to their semantic meaning. We ask different subjects to work independently. Majority rule is used to resolve different opinions when they arise. Details of part state annotation are presented in the supplementary file.

To verify the generality of collected part states, we also ask 8 subjects to label part states (without part segmentation) in ImageNet. We found that our current part-states definitions can cover 97.1% cases in the ImageNet dataset (in the labeled 15 categories) which our dataset collects a large variety of natural scenes. Additionally, our proposed description generation scheme is scalable, which can easily expand the semantic space with more data available.

4. Iterative Part-state Inference Network

In this section, we present the Iterative Part-state Inference Network (ISIN) to simultaneously predict part states and part segmentation on object images. The network operates on a novel RGB-S image format. In the training phase, we learn a model from images with annotation. In the testing phase, part segments and part states on an RGB image without annotation are predicted. We learn different models

for different categories independently. Mathematically, we use a binary variable to indicate whether a particular part state exists (i.e., 1 for exist, 0 otherwise). We concatenate all the binary variables into a "part state vector". If a part is missing in an image, the vector is a zero vector. In the following, we will first introduce the part network block, and then the RGB-S image and finally detail the iterative part-state inference network.

4.1. Part Localization Network

This network aims at localizing in a pixel-wise manner part regions given an object image. Therefore, the problem can be modeled as semantic segmentation. We denote the part segmentation networks with input 6 and 3 channels respectively as Part-6 network and Part-3 network. Specifically, the image input to the part network is of dimension $W \times H \times F$, where $W \times H$ are the spatial dimensions of the object image and F is the number of channels, which can be 3 or 6. The segmentation solver outputs a $W \times H \times C$ volume, where C is the number of part categories. The ith layer of the volume is a pixel-wise probability map of the ith part class. We adopt an end-to-end deconvolution network [17], which is one of the state-of-the-art semantic segmentation solvers, to segment the parts.

4.2. Iterative RGB-S Representation

We stack the input RGB image I (resized to 224×224) and its part-segmented image S (with size 224×224) to form an RGB-S image (see examples in Figure 4). We denote the RGB-S image as u (with size $224 \times 224 \times 6$). S is a 3-channel image to indicate the parts in distinct colors. Recall that the part network outputs C probability maps for C parts. We linearly map this volume into a 3-channel color image with a fixed mapping matrix $\mathcal{M}$ in $\mathbb{R}^{k \times 3}$, where k is the number of parts. We uniformly sample k colors from RGB color space to be the rows of M, where k is the number of object part. In this way, sampled colors are distinguishable from each other. Using M, we map in a pixel-wise manner k-channels probability volume to a 3-channel one. Each part will be marked as a sampled color in the mapped S image (named as part image).

In the beginning, the initial part image S_1 is obtained by training a Part-3 network on RGB image . With the initial RGB-S image $u_1 = \{S_1, I\}$, we iteratively improve the image by implementing a Part-6 network $f(u; \Theta_f)$, which can receive a RGB-S input ($224 \times 224 \times 6$) where Θ_f is the network parameter.

In the ith step, the updating of u can be expressed as

$$S_{i-1} = \mathcal{M} \cdot f(u_{i-1}; \Theta_f) \qquad (1)$$
$$u_i = \{S_{i-1}; I\} \qquad (2)$$

where u_i is the RGB-S image in the ith iteration, and $\mathcal{M} \cdot$ is a linear mapping operator over f.

The RGB-S representation encodes both local parts and holistic object information: the target parts are highlighted with pixel-wise part shape to let the network look into a particular part region, while the global object appearance is revealed in the RGB image. Mapping the segmentation score volume into 3 channels reduces the computation while conveying sufficient part information. As shown in Figure 4, we can visually distinguish different parts in the S image.

4.3. Iterative Joint Learning of Part Segmentation and Part State

In [3], employing feedback in building predictors is shown to be effective in handling complex structure (e.g. in pose estimation), which echoes human visual system where feedback connections are abounded [5]. Inspired by this, we employ an iterative scheme to jointly learn part segmentation and part state. Prediction error of the part state can be considered as a feedback of part segmentation, because good part segmentation can improve part state prediction. The part state will in turn help to guide the part segmentation (e.g., if the keyboard is not used, it should be shaped like a quadrilateral). Thus, we propose to iteratively refine part segmentation labeling under the guidance of part states which encode the knowledge of part appearance. A better part segmentation will in turn lead to improved part states as the iterations proceed.

Part-state Guidance Our part state vector is predicted given an RGB-S image. Denote $g(\cdot)$ as the state network whose input and output are respectively an RGB-S image and part state vector. Our problem can be considered as one of multi-class labeling, so we adopt the VGG network [22] to solve the problem.

In the ith step, denote the input as u_i and the output as a_i. Then, we have

$$a_i = g(u_i, \Theta_g) = g(\{\mathcal{M} \cdot f(u_{i-1}, \Theta_f); I\}, \Theta_g) \qquad (3)$$

where Θ_g is the network parameter.

Objective Function In the ith iteration, we jointly minimize the two tasks. Given u_{i-1}, the objective function $\min_{\{\Theta_g, \Theta_f\}} G_i(\Theta_g, \Theta_f)$ is

$$\sum_{j=0}^{N} \{l[g((f(u_{i-1}^j), \Theta_f), \Theta_g), a_{gt}^j] \\ + \lambda l[f(u_{i-1}^j, \Theta_f), s_{gt}^j]\} \qquad (4)$$

where N is the total number of training samples, u_i^j, a_{gt}^j and s_{gt}^j are respectively the RGB-S image, ground truth part state vector and part segments of the jth sample; $\lambda = 0.2$ is a hyper-parameter that was obtained through grid search
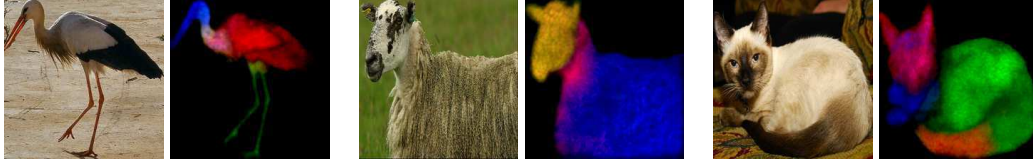
Figure 4: RGB-S image examples. In each pair, the left is the RGB image and right is S image.

for maximizing the performance on the validation set. The function $l[\cdot]$ measures the distance in the form of a soft-max loss error. We iteratively train the model. The stopping criterion can either be the loss error being smaller than a certain threshold, or the iteration number exceeding a maximum number $M = 12$. Experimental results in the following show that part state prediction is progressively refined thanks to the improvement of the part segmentation. This is due to the fact that the network parameter of the previous iteration makes a good initialization for the network training in the current iteration. We optimize the cost function Eq. (4) using stochastic gradient descent (SGD). Typically the iterative optimization converges in 6–8 iterations.

Given an image, in the testing phase, we iteratively compute the learned $f(\cdot)$ to produce a part segment to form the RGB-S image. The number of iterations is the same as in the training phase. In the last step, part state vector is predicted based on the final RGB-S image by computing $g(\cdot)$.

The Unfolded Architecture We find that our iterative framework can be unfolded into a sequential architecture as shown in Figure 6. This unfolded architecture, in fact, looks similar to a recurrent neural network (RNN) [1, 15]. However, our problem is significantly different from those solved by conventional recurrent neural network and hence our resulting architecture is fundamentally different. Firstly, our data is not sequential and is not well-suited for the RNN. Secondly, according to the unfolded model in Figure 6, our S is analogous to the hidden units of RNN, but in RNN, the hidden units are free in the learning process, while we impose constraints on S to encourage part segment formation.

Nevertheless, for the sake of comparison, we also train the unfolded architecture. Directly training the sequential objective function will lead to training a very deep model which is very time-consuming. So, we train a sub-sequence

Figure 5: The shape of a part segmentation mask is related with its part state: (a) and (b) "keyboard is being used", (c) and (d) "keyboard is not being used". In (a) and (b), keyboards are occluded by human hands, thus the occluded area is excluded from the part segmentation mask.

iteratively. We minimize the objective function involving the error sum from k^{th} to $(k+h)^{th}$ iterations. The objective function can be expressed as

$$\min_{\Theta_g, \Theta_f} \sum_{i=k}^{h+k} G_i(\Theta_g, \Theta_f) \qquad (5)$$

The optimization result u_{h+k} will be used to train the next round which minimizes the error sum from $k + h + 1$ to $k + 2h$.

Experimentally, the significant extra computation for this setup only marginally improves the performance comparing with our proposed iterative framework. One possible explanation is that the iterative architecture is already a good approximation of this unfolded architecture.

5. Experiments

In this section, we first introduce the evaluation metric and baseline methods for comparison, followed by presenting a discussion. Qualitative experiments will then be described. Finally, we apply our part state method on visual relationship recognition.

5.1. Evaluation Metric

Our task consists of detecting part states with corrected localization which is analogous to object detection. We revisit the typical evaluation metric of object detection: if the ratio intersection over union (IoU) between the predicted object box and the ground truth bounding box is larger than 0.5, and that the confidence score in the ground truth category is larger than a threshold, then we say this is a correct detection. By varying the thresholds, we can produce different precisions under different recalls. The average precisions (AP) is used to evaluate the performance.

We adopt this metric to produce a mAP measure that is reported by the mean APs over all part categories. The only difference is that the IoU we compute here is based on pixel-wise segments rather than bounding boxes. We do not use bounding box since a large number of non-compact parts simply cannot be accurately delineated using a bounding box.

5.2. Baseline Methods

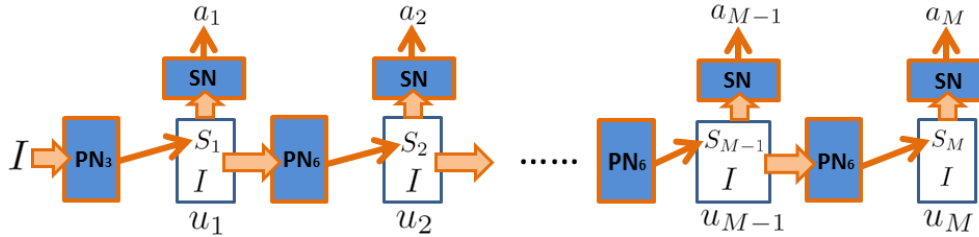The following baseline methods are compared:

Figure 6: The unfolded network from iterations 1 to M, where PN_6 is Part-6 network on the RGB-S image and SN is the state network. PN_3 is Part-3 network on the RGB image.

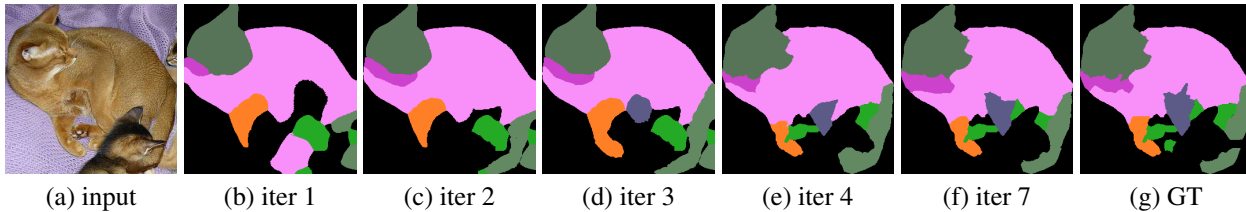| (a) input | (b) iter 1 | (c) iter 2 | (d) iter 3 | (e) iter 4 | (f) iter 7 | (g) GT |

Figure 7: The segmentation map is improved while iterations are in progress. The iteration converges within $6 \sim 8$ "GT" is ground truth.

Baseline 1: Global-RGB model We directly train a state network (using the VGG architecture [22]) on the RGB image to predict the part state vector. The learned model is named as global-RGB model. In the testing phase, we produce the part state vector on the input RGB image directly.

Baseline 2: Local-RGB model We predict part states in local part regions. We use the part network (training and testing) on the RGB image to localize parts. Then, the regions tightly bounding the parts are extracted, and we implement the VGG network on them to predict the part state vectors. The learned model is named as local-RGB model.

Baseline 3: Global+Local RGB model We combine the previous two baseline models. The parameters of the last layer of the local-RGB and global-RGB model are respectively extracted and then concatenated to form a vector. Binary SVMs are trained on the vectors to predict the elements of the part state vector.

5.3. Quantitative Evaluation

Table 1 compares the results of the above baseline methods and our method under different settings:

Setting 1: Implementing one iteration only in the training. That is, the S image is not iteratively updated.

Setting 2: Learning the model to use our iterative state-part guided network as described in section 4.

Setting 3: Training the unfolded architecture model with Eq. (5) (using subsequence length $h = 3$).

The results of baseline 1 and baseline 2 indicate that it is insufficient to model only globally (whole object image) or locally (part regions): to effectively perceive a part we must consider both the local part appearance and the object context. For baseline 3, although local and global information are considered, they are not jointly learned, which explains the performance drop of 7% in mAP when compared with our method. In contrast, our RGB-S image format can effectively derive local part regions in the pixel level (S image), while providing the object appearance and its relationship among different parts (RGB image). Furthermore, our experiments verify that the iterative scheme (*setting 2*) outperforms the non-iterative solver (*setting 1*) by 5%. If we train on the unfolded architecture Eq. (5), the improvement is very minor at 0.4%. A possible explanation is that iterative training (*setting 2*) is already a good approximation. The drawback of the unfolded architecture (*setting 3*), however, is the large computation cost. In short, we recommend *setting 2* for solving this problem, which is a good balance between effectiveness and computation cost.

5.4. Discussion

Iteration The results in Table 1 shows that the iterative method on RGB-S image outperforms the non-iterative method. Table 2 tabulates the detail of iterations, where the mAP and segmentation accuracy in each iteration are shown. An example of part segmentation during iterations is shown in Figure 7. Both quantitative results and qualitative results (below) demonstrate that our iterative scheme can indeed improve the performance. The solver is always nicely converge within $6 \sim 8$ round iterations.

	aero	bike	bird	bottle	bus	car	cat	cow	dog	horse	mbike	person	plant	sheep	train	ave
B1	26.7	10.9	27.0	24.7	30.2	28.0	31.8	24.2	27.3	24.2	28.8	28.9	18.0	26.6	26.4	25.7
B2	29.6	12.5	31.7	26.2	30.8	28.2	31.8	25.0	29.6	26.8	38.4	29.1	23.5	27.8	25.9	27.8
B3	32.4	14.6	34.5	35.4	38.8	39.8	35.9	31.7	32.8	28.8	36.3	32.3	25.1	33.4	32.7	32.3
S1	44.9	15.1	44.3	39.9	**56.9**	49.4	44.8	38.5	45.8	37.0	43.0	45.6	30.3	42.1	33.1	40.7
S2	**52.1**	**20.1**	50.3	39.9	53.5	58.2	44.3	**45.2**	**53.1**	38.0	53.5	52.0	**37.9**	**52.4**	**47.0**	46.5
S3	45.2	19.8	**50.7**	**50.1**	55.2	**59.8**	**52.9**	42.2	42.6	**43.2**	**56.8**	**55.2**	37.5	46.0	46.3	**46.9**
GT Seg	66.7	25.7	66.7	60.2	75.3	70.6	69.6	57.7	61.0	57.7	72.0	68.9	50.4	61.4	58.5	61.5

Table 1: The mean average precision (mAP) for different object categories on the PASCAL VOC 2010 part state dataset. Baselines 1,2 and 3 are respectively denoted as **B1**, **B2**, and **B3**. Settings 1,2 and 3 are respectively denoted as **S1**, **S2**, and **S3**. "GT Seg" means we use ground truth segment image as the part-segmented image (S).
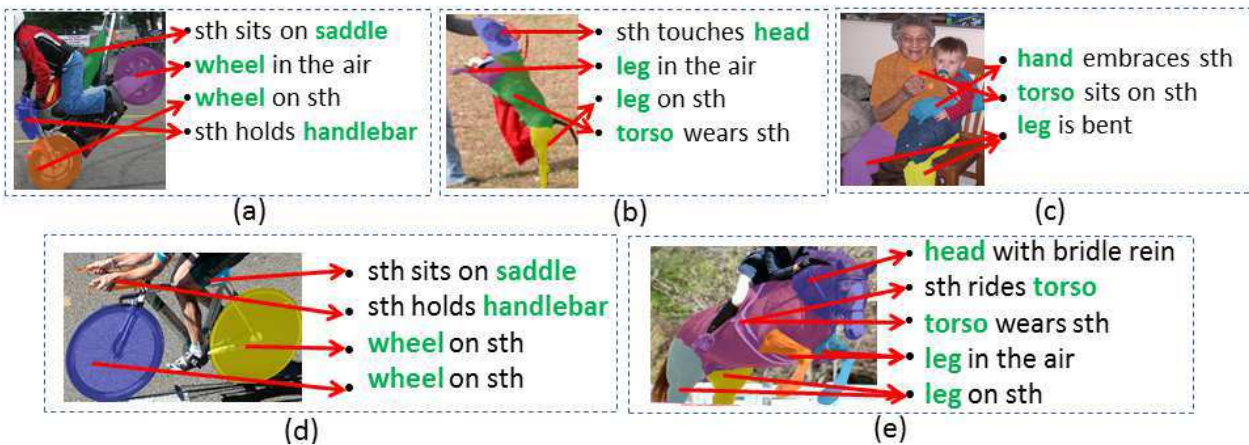

Figure 8: Representative results where the detected part segments and part state on them are illustrated.

Influence of Segmentation We study the case when perfect segmentation is available in RGB-S image, which will effectively eliminate the influence of segmentation error. Table 1 shows the result in the 'Seg GT' row. We find that even given perfect pixel-wise part localization, we may still not be able to perfectly predict the correct part states. The possible explanation is that the semantic meaning conveyed by the pertinent parts are beyond simple part shape patterns.

5.5. Qualitative Experiments

Figure 8 shows representative part state predictions. We find that the parts are well segmented and the part states are quite accurately predicted, although we observe some imperfect part segmentation (see the torso in Figure 8(c)).

5.6. Relationship Prediction

We apply our part state method on visual relationship recognition. We use the visual relationship dataset [14] which includes object-pairs relationship annotation, such as "person holds cup". We select the relationships where each object-pair contains at least one object in our 15 categories. After manual refinement, we have 6429 object-part relation-

ships (5000 for training, 1425 for testing) which include 31 predicate types, such as "hold", "push" and "under". We found giving two object names alone is limited to infer their relationship. We should look into part level to further judge.

Solution with Part State. The part state vectors capture rich and explicit relationship information. For instance, if the part state bin for "hand holds something" is 1 and something is detected as a cell phone, it is straightforward to derive the relationship is "person holds phone". Therefore, we can extract part state vectors as feature vectors. To unify the feature length, we pad zeros for vectors with length smaller than 72, where 72 is the maximum vector length among the 15 object categories. Then, SVM is applied to the concatenated part state vector of two objects. We combine our SVM score with the score of [14] by simply averaging them as final relationship prediction score.

Baselines. For *baseline 1*, we use visual phrase recognition [19] to classify the relationships. For *baseline 2*, we follow [22] to jointly learn object and relationship in a unified CNN. For *baseline 3*, we adopt the conventional computer vision scheme: concatenate the CNN feature from

Iteration	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	#12
mIoU	57.6	60.8	62.0	63.6	65.4	67.1	70.1	70.8	71.2	71.3	71.2	71.3
mAP	40.7	41.6	42.5	43.3	44.0	44.8	45.6	45.9	46.2	46.3	46.3	46.5

Table 2: Average part state mAP (second row) and segmentation accuracy (mean IoU) of part categories (first row) of as iterations are proceeding.

	Phrase Det.		Relationship Det.		Predicate Det.	
	R@100	R@50	R@100	R@50	R@100	R@50
[19]	0.05	0.04	-	-	1.82	0.92
[22]	0.09	0.06	0.08	0.07	1.94	1.36
B3	9.42	8.91	9.04	7.85	36.15	36.15
[14]	16.32	15.80	13.01	12.48	44.19	44.19
Ours	**25.37**	**24.80**	**26.13**	**24.48**	**53.50**	**53.50**

Table 3: Results of visual relationship detection. Note that the dataset we use is different from the one used in [14]. Here, [19], [22] and B3 respectively refer to baseline 1, 2 and 3. We use the relationships with at least one object in our 15 categories. R@100 and R@50 are respectively the abbreviations for Recall @ 100 and Recall @ 50. Note that in predicate det., we are predicting multiple predicates per image (one between every pair of objects) and hence R@100 is less than 1.

BL 1: person **contains** cellphone
BL 2: person **under** cellphone
BL 3: person **watches** cellphone
Lu *et al.*: person **watches** cellphone
Ours: person holds cellphone

part-state: hand holds sth

Figure 9: An example of relationship prediction. BL1 is baseline 1 (visual phrase), BL2 is baseline 2(Joint CNN), BL3 is baseline 3 (vision-language prior) and Lu *et al.* refers to [14]. "GT" is the ground truth. The part state at hand is correctly detected by the proposed method.

VGG [22] and the word-vector [16] (language prior) of two objects to form a baseline feature. Then, we use SVM to classify the relationship types. Approach of [14] is taken as the *fourth baseline*.

Results. The result is tabulated in Table 3, where we follow the convention recall @ 100 and recall @ 50 in [14]. Recall @ **x** computes the fraction of times the correct relationship is predicted in the top **x** confident relationship predictions. The first and second baselines do not have a language prior. For baseline 3 and [14], the language prior improves performance. However they preform at the holistic object level. In contrast, our method which incorporates part level information significantly advances 12 mAP the detection performance.

Analysis. Object relationship is a higher level concept than holistic objects, and our part state encodes richer information such as interaction, affordance and functionality. Our method shows good promise and we believe more powerful tools based on part states can be explored. Figure 9 demonstrates an example: given the complex concept of

"hold" the holistic object-level appearance is not sufficient, so we should look into the key part region – hand.

6. Conclusion and Future Work

We have presented *part state* to tokenize the semantic space of object parts and explore richer semantic information for image understanding. With the proposed iterative part-state inference Network operating on RGB-S representation, we can iteratively improve part state prediction. Extensive experiments have demonstrated the proposed method outperforms various baseline methods. Our part state can be applied to object relationship prediction and very promising results are obtained.

One limitation is that the model we trained is not class-agnostic. That is, we cannot use a unified model for all categories. The difficulty in training a unified, class-agnostic model stems from the fact that the output part states for different categories vary largely among each other. We find about 8% mAP performance drop if the model is trained without category consideration. Future work is therefore to learn a unified model without significant performance degradation in comparison to independently-learned category-specific models. The other limitation is that we still have no theoretical stopping criterion on iterations. We will also perform a principled study of the unfolded model to explain its incremental improvement in comparison to the iterative model.

References

[1] P. J. Angeline, G. M. Saunders, and J. B. Pollack. An evolutionary algorithm that constructs recurrent neural networks. *IEEE Transactions on Neural Networks*, 5(1):54–65, 1994. 5

[2] L. Bourdev and J. Malik. Poselets: Body part detectors trained using 3d human pose annotations. In *CVPR*, 2009. 3

[3] J. Carreira, P. Agrawal, K. Fragkiadaki, and J. Malik. Human pose estimation with iterative error feedback. *arXiv preprint arXiv:1507.06550*, 2015. 4

[4] 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. 3

[5] D. J. Felleman and D. C. Van Essen. Distributed hierarchical processing in the primate cerebral cortex. *Cerebral cortex*, 1(1):1–47, 1991. 4

[6] P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *CVPR*, 2008. 2

[7] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014. 3

[8] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015. 3

[9] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *CVPR*, 2015. 3

[10] R. Keen. The development of problem solving in young children: A critical cognitive skill. *Annual review of psychology*, 62:1–21, 2011. 2

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 3

[12] B. Li, T. Wu, C. Xiong, and S.-C. Zhu. Recognizing car fluents from video. *arXiv preprint arXiv:1603.08067*, 2016. 1

[13] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 3

[14] C. Lu, R. Krishna, M. Bernstein, and L. Fei-Fei. Visual relationship detection with language priors. In *ECCV*, 2016. 7, 8

[15] L. Medsker and L. Jain. Recurrent neural networks. *Design and Applications*, 2001. 5

[16] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, 2013. 8

[17] H. Noh, S. Hong, and B. Han. Learning deconvolution network for semantic segmentation. In *CVPR*, 2015. 3, 4

[18] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, 2015. 3

[19] M. A. Sadeghi and A. Farhadi. Recognition using visual phrases. In *CVPR*, 2011. 7, 8

[20] T. Shu, M. Ryoo, and S.-C. Zhu. Learning social affordance for human-robot interaction. In *IJCAI 2016*. 1

[21] N. Shukla, C. Xiong, and S.-C. Zhu. A unified framework for humanrobot knowledge transfer. In *AAI Fall Symposium Series*, 2015. 1

[22] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 3, 4, 6, 7, 8

[23] J. J. Tompson, A. Jain, Y. LeCun, and C. Bregler. Joint training of a convolutional network and a graphical model for human pose estimation. In *NIPS*, 2014. 3

[24] A. Toshev and C. Szegedy. Deeppose: Human pose estimation via deep neural networks. In *CVPR*, 2014. 3

[25] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3156–3164, 2015. 2, 3

[26] J. Wang and A. L. Yuille. Semantic part segmentation using compositional model combining shape and appearance. In *CVPR*, 2015. 1, 3

[27] J. Yang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1794–1801. IEEE, 2009. 2