Learning Object Interactions and Descriptions for Semantic Image Segmentation

Guangrun Wang[1,2] Ping Luo[2,4] Liang Lin[1,3] Xiaogang Wang[1,4]

[1]Sun Yat-sen University [2]The Chinese University of Hong Kong [3]SenseTime Group (Limited)

[4]Shenzhen Key Lab of Comp. Vis. & Pat. Rec., Shenzhen Institutes of Advanced Technology, CAS, China

wanggrun@mail2.sysu.edu.cn pluo@ie.cuhk.edu.hk linliang@ieee.org xgwang@ee.cuhk.edu.hk

Abstract

Recent advanced deep convolutional networks (CNNs) achieved great successes in many computer vision tasks, because of their compelling learning complexity and the presences of large-scale labeled data. However, as obtaining per-pixel annotations is expensive, performances of CNNs in semantic image segmentation are not fully exploited. This work significantly increases segmentation accuracy of CNNs by learning from an Image Descriptions in the Wild (IDW) dataset. Unlike previous image captioning datasets, where captions were manually and densely annotated, images and their descriptions in IDW are automatically downloaded from Internet without any manual cleaning and refinement. An IDW-CNN is proposed to jointly train IDW and existing image segmentation dataset such as Pascal VOC 2012 (VOC). It has two appealing properties. First, knowledge from different datasets can be fully explored and transferred from each other to improve performance. Second, segmentation accuracy in VOC can be constantly increased when selecting more data from IDW. Extensive experiments demonstrate the effectiveness and scalability of IDW-CNN, which outperforms existing best-performing system by 12% on VOC12 test set.

1. Introduction

Performances of convolutional networks (CNNs) can be improved by increasing depths, number of parameters, and number of labeled training data. They achieved state-of-the-art results and even surpassed the performances of human experts in image recognition [6, 7, 30] and object detection [26, 21]. Nevertheless, since training data with per-pixel annotations are limited and difficult to obtain in semantic image segmentation, performance gain of CNNs by merely increasing its modeling complexity becomes marginal.

To address data limitation in image segmentation, this work proposes to jointly train CNN from two sources of data. One is a small set of images with per-pixel annotations, which are difficult to obtain such as VOC12. An image of 'human' and 'sheep' in VOC12 and its annotation are given in Fig.1 (b). The other one is a large set of images automatically downloaded from the Internet, using the categories of VOC12 as keywords. Each image is equipped with an Image Description in the Wild but without per-pixel annotation. This image set is abbreviated as IDW. Unlike existing image captioning dataset such as MS COCO [13], where captions are manually generated by satisfying some annotation rules, image descriptions in IDW are directly copied from the web pages, including news, blog, forum, and photography agency. Fig.1 (a) provides an example with 'human' and 'sheep' as keywords, where shows that raw description of IDW may contain unimportant details, missing details, and grammatical errors.

With VOC12 and IDW, a novel CNN structure namely

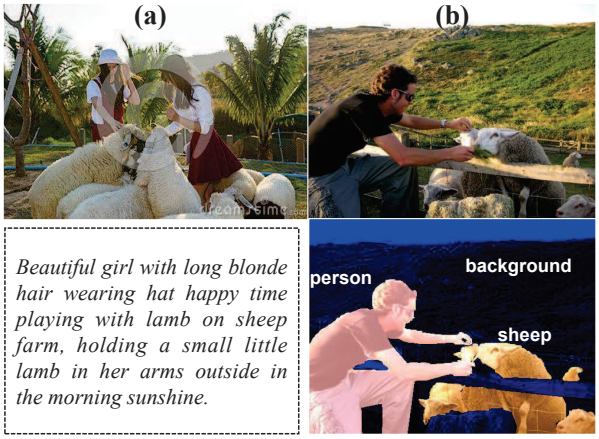


Figure 1: (a) visualizes an image in IDW and its raw description, searched by using 'sheep' and 'human' as keywords. We observe that the description contains unimportant details for object segmentation (*e.g.* 'long blonde hair wearing hat happy time'), missing important details (*e.g.* number of people and sheep), and grammatical errors. As a side-by-side comparison, (b) shows an image and its per-pixel annotation of VOC12.

The first two authors share first-authorship.

IDW-CNN and its training algorithm are carefully devised, where knowledge of these two datasets can be transferred from each other. Specifically, useful object interactions or relationships can be extracted from IDW, such as 'girl holding sheep' and 'girl playing with sheep', which are not encoded in the per-pixel labelmaps of VOC12. These object interactions can be transferred to VOC12 to improve segmentation accuracy. In addition, labelmaps in VOC12 that capture precise object locations and boundaries are able to improve the extractions of object interactions in IDW. These two purposes are formulated cooperatively in IDW-CNN and learned end-to-end. Extensive studies show that after training, accuracies of image segmentation and prediction of object interactions in both VOC12 and IDW are significantly increased. A more appealing property is that segmentation accuracy of VOC12 can be constantly improved, when adding more data to IDW. For instance, adopting 10 thousand images in IDW increases the accuracy of a state-of-the-art system, DeepLab-v2 [3], by 7.6% in VOC12 test set (from 74.2% to 81.8%). Adding another 10 thousand samples brings extra 3.37% improvement. Another 1.1% improvement can be achieved when introducing 20 thousand more samples. In general, IDW-CNN increases accuracy of DeepLab-v2 by 12% without any post-processing such as MRF/CRF smoothing [3].

This work has three main **contributions**. (1) This is the first attempt to show that image descriptions in the wild without manually cleaning and refinement are able to improve image segmentation. An IDW dataset containing more than 40 thousand images are constructed to demonstrate this result. (2) IDW-CNN is proposed to jointly learn from VOC12 and IDW. Knowledge from both datasets are fully explored and transferred from each other. Performances of segmentation and object interaction prediction in both datasets can be significantly improved. (3) IDW-CNN is capable of constantly improving segmentation accuracy, when more data are appended to IDW, showing its scalability and potential in large-scale applications.

1.1. Related Work

Supervised Image Segmentation CNNs achieved outstanding performances in semantic image segmentation. For instance, Long *et al.* [16] transformed fully-connected layers of CNN into fully convolutional layers (FCN), making accurate per-pixel classification possible by the contemporary CNN architectures that were pre-trained on ImageNet [27]. Since then, the combination of FCN and MRF/CRF attracts a lot of attention, and achieved great successes in semantic image segmentation [15, 2, 28, 34]. However, all works above use per-pixel annotations as full supervision, which are limited and hard to obtain.

Semi- and Weakly-supervised Image Segmentation Previous works [14, 24, 25, 23] tried to solve semantic image segmentation using only weak labels (*e.g.* image-level annotation), which are easier to attain but the problem is ill-posed and more challenging. Recent works [22, 4, 10, 8, 12] address this trade-off by combining both the weak and strong labels to reduce labeling efforts while improve segmentation performance. Supervision of these works are compared in Table 1, including per-pixel annotation, image-level annotation, bounding boxes (bbox), scribble, and language. Typically, the methods leverage both the pixel-level labels and weak labels (*e.g.* image and bbox) outperform the others.

Table 1: Comparisons of semi- and weakly-supervised image segmentation methods. Different approaches utilize different supervision as indicated by '√'. Different from the other methods that employed manually annotated labels, IDW-CNN learns from images and descriptions without any human intervention.

	Pixel	Img Tag	BBox	Scribble	Language
WSSL(weak)[22]		√			
WSSL(semi)[22]	√	√	√		
MIL-FCN[24]		√			
MIL-sppxl[25]		√			
CCNN[23]		√			
BoxSup[4]	√		√		
ScribbleSup[10]				√	
NLE [8]					√
DeepStruct[12]	√				√
IDW-CNN	√				√

For example, WSSL(semi) [22] improves accuracy of VOC12 *val* set from 62.5% to 65.1% by leveraging additional manually labeled bounding boxes and image-level tags. BoxSup [4] also benefits from bounding box annotations. NLE [8] and DeepStruct [12] employed language models in image segmentation, but addressed a task different from most of previous works, by parsing an image into structured regions according to a language expression. The key disparity between the above approaches and IDW-CNN is that previous works leveraged manual annotations but IDW-CNN does not. Extensive experiments show that IDW-CNN outperforms existing methods by a significantly large margin.

2. Learning Image Descriptions

Data Collection We construct an image description in the wild (IDW) dataset to improve the segmentation accuracy in VOC12. IDW is built with two stages, which can be easily generalized to different benchmarks other than VOC12. In the first stage, we prepare 21 prepositions and verbs that are frequently presented, such as 'hold', 'play with', 'hug', 'ride', and 'stand near', and 20 object categories from VOC12 such as 'person', 'cow', 'bike', 'sheep', and 'table'. Their combinations in terms of 'subject + verb/prep. + object' leads to $20 \times 21 \times 20 = 8400$ different phrases, such as 'person ride bike', 'person sit near

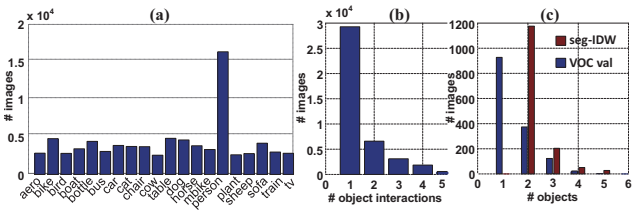


Figure 2: The statistics of IDW dataset.

bike', and 'person stand near bike'. However, the semantic meanings of most of these phrases are rarely presented in practice, for example 'cow hug sheep'. After removing meaningless phrases, we collect hundreds of meaningful phrases. In the second stage, these phrases are used as key words to search images and their surrounding texts from the Internet. We further discard the invalid phrases, such as 'person ride cow', if the number of their retrieved images is smaller than 150 to prevent rare cases or outliers, which may lead to over-fitting in training. As a result, we have 59 valid phrases. Finally, IDW has $41,421$ images and descriptions. Fig.2 (a) plots the number of images in IDW with respect to each object category in VOC12. This histogram reveals the image distribution of these objects in real world without any manually cleaning and refinement.

Image Description Representation Each image description is automatically turned into a parse tree, where we select useful objects (*e.g.* nouns) and actions (*e.g.* verbs) as supervisions during training. Each configuration of two objects and the action between them can be considered as an object interaction, which is valuable information for image segmentation but it is not presented in the labelmaps of VOC12.

Here, we use the Stanford Parser [29] to parse image descriptions and produce constituency trees, which are two-way trees with each word in a sentence as a leaf node, as shown in Fig.3(a). Constituency trees from the Stanford Parser still contains irrelevant words that do neither describe object categories nor interactions (*e.g.* adjectives). Therefore, we need to convert constituency trees into semantic trees, which only contains objects and their interactions. The conversion process generally involves three steps. Given a constituency tree in (a), we first filter the leaf nodes by their part-of-speech, preserving only *nouns* as object candidates, and *verbs* and *prepositions* as action candidates. Second, *nouns* are converted to objects. We use the lexical relation data in WordNet [19] to unify the synonyms. Those *nouns* that do not belong to the 20 object categories will be removed from the tree. Third, *verbs* should also be recognized and refined. We map the *verbs* to the defined 21 actions using word2vec [18]. When the mapping similarity

Images and descriptions are downloaded from photography agency such as www.dreamstime.com.

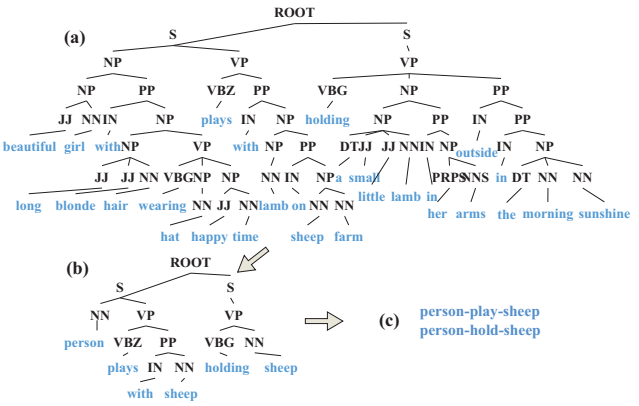


Figure 3: An illustration of image description representation. (a) is the constituency tree generated by language parser. (b) is the constituency tree after POS tag filtering. (c) presents the object interactions.

is smaller than a threshold, the *verbs* will be categorized into an additional action class, *i.e.* 'unknown'. Step 1 to 3 are shown in Fig.3. Finally, we extract the object interactions from the semantic tree through the nodes. An example of description parsing is illustrated in (b), where the principal component message 'girl plays with lamb, holding lamb' is first filtered out of the description, and then is further transferred into 'person plays with sheep, holding sheep'.

After parsing all image descriptions in IDW, we obtain $62,100$ object interactions in total. Fig.2 (b) summaries the number of images with respect to the number of interactions, showing that each image has 1.5 interactions on average. Different from existing datasets such as Visual Genome [9] that each image is manually and densely labeled with 17.68 relationships per image, the construction of IDW has no manual intervention and has extremely low expense compared to previous datasets. By partitioning IDW into different subsets, we show that IDW-CNN is able to progressively improve accuracy of VOC12 when training with more subsets of IDW.

To evaluate the generalization capacity of IDW-CNN, three test sets are constructed. First, we randomly choose $1,440$ images from IDW as a test set of object interaction prediction, denoted as int-IDW. These images are not utilized in training. Second, we annotate the per-pixel labelmap for each image in int-IDW, resulting in a segmentation test set, denoted as seg-IDW. Fig.2 (c) plots the number of images with respect to the number of objects, where seg-IDW is compared to the validation set of VOC12, indicating that seg-IDW is more challenging than VOC12 in terms of the object diversity in each image. Third, another interesting evaluation is a zero-shot test set denoted as zero-IDW, which includes $1,000$ images of unseen object interactions. For instance, the image of 'person ride cow'

is a rare case (*e.g.* in bullfight) and is not appeared in training. This evaluates that IDW-CNN is able to generalize to unseen object interactions.

2.1. Network Overview

Fig.4 (a) illustrates the diagram of IDW-CNN, which has three important parts, including a ResNet-101 network for feature extraction, a network stream for image segmentation (denoted as 'Seg-stream'), and another stream for object interaction (denoted as 'Int-stream'). They are discussed as below.

Feature Extraction IDW-CNN employs DeepLab-v2 [3] as a building block for feature extraction. It is a recent advanced image segmentation system, incorporating ResNet-101, multi-scale fusion, and CRF smoothing into a unified framework. To identify the usefulness of IDW dataset, IDW-CNN only inherits ResNet-101 from DeepLab-v2, yet removing the other components such as multi-scale fusion and CRF in DeepLab-v2. Given an image I, ResNet-101 produces features of 2048 channels. The size of each channel is 45×45.

Seg-stream As shown in Fig.4 (a), the above features are employed by a convolutional layer to predict segmentation labelmap (denoted as $\widetilde{I}^s$), the size of which is $21 \times 45 \times 45$. Each channel indicates the possibility of an object category presented in image I. The final prediction I^s is produced by refining $\widetilde{I}^s$ using object interaction. The component of refinement will be introduced below.

Int-stream This stream has three stages. In the first stage, we reduce the number of feature channels from 2048 to 512 by a convolutional layer, denoted as h, so as to decrease computations for the subsequent stages. Then, we produce a set of 21 object feature maps, denoted as $\{h_i^m\}$ where the subscript $i \in \mathcal{C}$ and $\mathcal{C} = \{\text{person}, \text{cow}, ..., \text{bkg}\}$[21]. Each h_i^m is obtained by preforming the elementwise product ("$\otimes$") between h and each channel of $\widetilde{I}^s$, which represents a mask. Therefore, each $h_i^m \in \mathbb{R}^{512\times45\times45}$ represents the masked features of the i-th object class. Examples of h^m_{person} and h^m_{bike} in an image are given in Fig. 5 (a) and (b).

In the second stage, each h_i^m is utilized as input to train a corresponding object subnet, which outputs a probability characterizing whether object i is presented in image I. Thus, as shown in orange in Fig.4 (a), we have 21 object subnets, which have the same network structures but their parameters are not shared. (b) visualizes this structure in orange, where h_i^m is forwarded to one convolution, one max pooling, and one full-connection layer. Overall, the second stage determines which objects are appeared in I.

In the third stage, we train 22 action subnets as outlined in blue, each of which predicts the action between two appeared objects. Similarly, these subnets have the same architectures but their parameters are not shared. As shown in (b), structure of the action subnet is analogous to that of the object subnet, except an elementwise sum ("$\oplus$") in the input (in blue). For instance, if both 'person' and 'bike' are presented in I, the combination of their features, $h^m_{\text{person}} \oplus h^m_{\text{bike}} \in \mathbb{R}^{512\times45\times45}$, is propagated to all action subnets. Then, the largest response is more likely to be produced by one of the following action subnets, 'ride', 'sit near', and 'stand near', to determine the true action between these two objects. The combination of features is performed in object pair selection as introduced below.

Object-Pair Selection (OPS) As shown in purple in Fig.4 (a), OPS is an important component in Int-stream, which merges features of the presented objects. For example, if object subnets of 'person', 'bike', and 'car' have high responses, each pair of features among h^m_{person}, h^m_{bike}, and h^m_{car} are summed together elementwisely, resulting in three combined features denoted as $h^m_{\text{person+bike}}$, $h^m_{\text{person+car}}$, and $h^m_{\text{bike+car}}$. An example of $h^m_{\text{person+bike}}$ is plotted in Fig.5 (c). To resolve the action between each pair of objects, each merged feature is then forwarded to all 22 action subnets as discussed above. An appealing property of OPS is that the number of object interactions of different images can be different, adaptively determining by the cooperation between two groups of object and action subnets.

Refinement This is an essential component in Seg-stream to improve segmentation accuracy. Recall that the i-th object subnet produces a score (probability), denoted as l_i^o in Fig.4(a), indicating how likely object i is appeared in image I. So, we concatenate all 21 scores as a vector $\mathbf{l^o} \in \mathbb{R}^{21\times1}$ and treat it as a filter to refine the segmentation map $\widetilde{I}^s$ using convolution. We have $I^s = \text{conv}(\widetilde{I}^s, \mathbf{l^o})$.

3. Training Approach

IDW-CNN jointly trains images from IDW and VOC12, by using back-propagation (BP) with stochastic gradient descent (SGD). Each image in IDW contains object interactions but without labelmap, whilst each image in VOC12 has labelmap but no interactions. Therefore, IDW-CNN posses a large challenge of missing labels. Unlike previous multitask deep models [31, 33] that ignore the gradients of an unlabeled sample in the training stage, IDW-CNN estimates a pseudo label for each sample and treats it as ground truth in BP. Experiments show that this process is important and improves performance. Here, we discuss the backward propagations of two streams, with respect to two kinds of data respectively.

Backwards of Seg-stream As shown in the first two red arrows of Fig.4 (a), seg-stream has two identical softmax loss functions. One minimizes the per-pixel discrepancy between a ground truth labelmap $\overline{I}^s$ and $\widetilde{I}^s$, whilst the other involves I^s. Both loss functions are indispensable

[21] These represent 21 action items and another one subnet indicates no action is performed between objects.

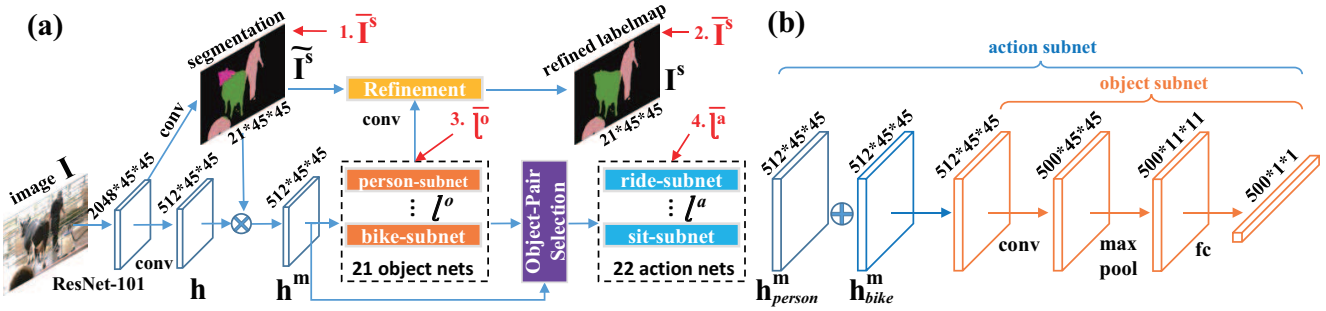
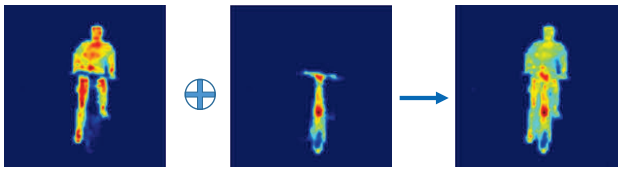


Figure 4: (a) illustrates the diagram of IDW-CNN, which has two streams. Given an image I, the first stream predicts its segmentation labelmap I^s and the second stream estimates its object interactions. The second stream contains two groups of sub-networks, where 21 object subnets recognize object classes appeared in the object interactions, while 22 action subnets predict action items between them. Each subnet has the same network structure as shown in (b), but are learned to achieve different goals. Thus, their parameters are not shared. 'conv', 'max pool', and 'fc' indicate convolution, max pooling, and full-connection respectively. IDW-CNN has four groups of losses functions as marked by 1.-4. in red.



(a) (b) (c)

Figure 5: Combined feature of *person* and *bicycle*. The features of *person* h^m_{person} and *bicycle* h^m_{bike} are visualized in (a) and (b), respectively. Then the combined feature $h^m_{\text{person+bike}}$ are the element-wise summation of h^m_{person} and h^m_{bike}, visualized in (c).

in seg-stream. The first one learns to update the $2048 \times 45 \times 45$ features of ResNet-101. Besides these features, the second one also updates 21 object subnets, improving object categorizations.

In the following, we employ subscripts 'voc' and 'idw' to distinguish the images and labels from each dataset respectively. In particular, for an image in *VOC*, the gradients of these two losses are calculated as in conventional BP, since the ground truth labelmap $\bar{I}^s_{\text{voc}}$ is available. However, given an image in *IDW*, as $\bar{I}^s_{\text{idw}}$ is unavailable, only the first loss is activated. We estimate a latent $\bar{I}^s_{\text{idw}}$ as 'pseudo ground truth' by combining the predicted segmentation map, $\tilde{I}^s_{\text{idw}}$, and the predicted object labels, $\mathbf{l}^o_{\text{idw}}$. Intuitively, to attain $\bar{I}^s_{\text{idw}}$, we zero those regions presented in $\tilde{I}^s_{\text{idw}}$ but their corresponding object labels are absent in $\mathbf{l}^o_{\text{idw}}$.

Backwards of Int-stream As illustrated in the third and forth red arrows of Fig. 4(a), int-stream consists of two groups of loss functions. In the first group, each object subnet is trained by a 1-of-2 softmax loss to determine if the specific object appeared in an image. In the second group, each action subnet produces a response, forming totally 22 responses. Then the entire action nets optimize a 1-of-22 softmax loss, the largest response represents the true action between these two objects. Here, we introduce BP for two datasets respectively. For an image in *IDW*, as both ground truth labels of objects and actions, $\bar{\mathbf{l}}^o_{\text{idw}}$ and $\bar{\mathbf{l}}^a_{\text{idw}}$, are available, gradients can be simply obtained by BP.

For an image in *VOC*, ground truth of the presence of an object, $\bar{\mathbf{l}}^o_{\text{voc}}$, can be easily determined from the labelmaps $\bar{I}^s_{\text{voc}}$. However, as the ground truth actions, $\bar{\mathbf{l}}^a_{\text{voc}}$, between objects are not available, they need to be inferred in the learning stage. For example, if 'bike' and 'person' are presented in I^s_{voc}, there are four different possible actions between them such as 'sit', 'stand', 'ride', and 'sleep'. In our implementation, we obtain a prior distribution with respect to actions between each pair of objects. For 'bike' and 'person', this prior produces high probabilities over the above four actions and low probabilities over the others. In the training stage, the loss function offers low penalty if the predicted action is among one the above, otherwise provides high penalty.

Implementation Details As mentioned in Sec.2.1, IDW-CNN employs ResNet-101 as building block, where the parameters are initialized by classifying one thousand image classes in ImageNet. The other parameters in IDW-CNN are initialized by sampling from a normal distribution. IDW-CNN is trained in an incremental manner with three stages.

First, all losses are deactivated except the first one as marked by '1.' in red of Fig. 4(a). In this stage, MS COCO [13] is used to train the model following [3], in order to adapt features from image classification to image segmentation. Second, three losses except the fourth one are optimized by using VOC training set to improve segmentation. Third, parameters are jointly fine-tuned by all loss functions to transfer knowledge between VOC and IDW.

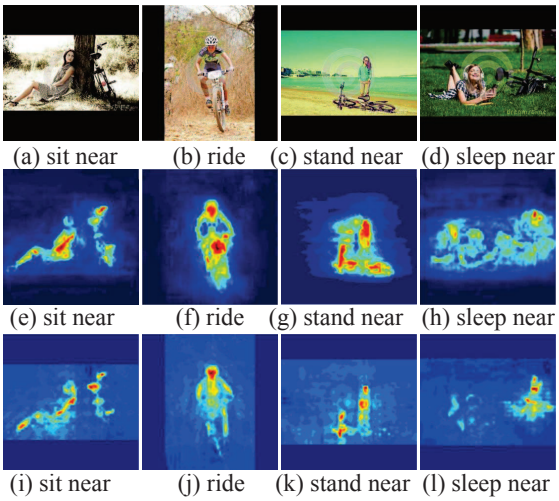


Figure 6: Visualization of interaction feature maps, where the verbs here denotes the interactions between 'person' and 'bike'. This results indicate that learning image description benefits image segmentation.

4. Experiments

We evaluate IDW-CNN in two aspects. Sec.4.1 conducts an extensive ablation study, including effectiveness of region pair selection, scalability, and object interaction prediction. Sec.4.2 compares segmentation accuracy of IDW-CNN with those of the state-of-the-art methods. To identify the usefulness of IDW, for all experiments, IDW-CNN employs ResNet-101 of DeepLab-v2 as backbone network, yet removing any pre- and post-processing such as multi-scale fusion and CRF. In this case, ResNet-101 achieves 74.15% accuracy compared to 79.7% of the full DeepLab-v2 model.

4.1. Ablation Study

Effectiveness of Object-Pair Selection (OPS) We compare performances of IDW-CNN with and without OPS. The latter method is abbreviated as 'IDW-CNN w/o OPS', which turns into a multi-task model, such that the entire shared features ($2048\times45\times45$ as in Fig.4) are directly utilized to predict both segmentation map and object interaction, without OPS as the full model did. We consider two variants of 'IDW-CNN w/o OPS'. The first one directly trains $20 + 22 + 20 = 62$ subnets, indicating two objects (20 categories each) and 22 actions. This is similar to [17], denoted as 'IDW-CNN w/o OPS-1'. The second one trains 59 subnets, corresponding to 59 valid object interactions, denoted as 'IDW-CNN w/o OPS-2'.

Segmentation accuracies on VOC12 test and seg-IDW are reported in Table 2 and 3 respectively, showing that performances drop 7.1% and 6.5% when removing OPS, which is a key to the success of IDW-CNN. It is worth noting that 'IDW-CNN w/o OPS-1' still outperforms ResNet-101 by 5% on VOC12 test, demonstrating the usefulness of IDW dataset. As an example, Fig.6 visualizes features of $h^m_{\text{bike+person}}$, which are the inputs to the action subnets. They help refine the predicted labelmaps. The first row shows the images. The second and third rows show the features of IDW-CNN and IDW-CNN w/o OPS-1 respectively. Good features for segmentation should have high responses on both objects. Intuitively, OPS learns discriminative features for 'person' and 'bike', therefore improving their segmentation performance. Fig.7 exhibits the effectiveness of action subnets. With OPS, each action subnet is able to identify informative region of a specific action. For instance, given the same image of two 'person' and a 'horse', both 'ride-subnet' and 'stand near-subnet' correctly identify the 'person' who is involved in the corresponding action. IDW captures this information, which is missing in VOC12.

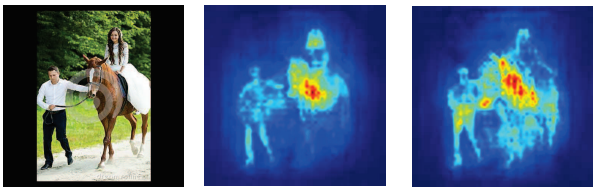


Figure 7: Visualizations of features in action subnets.

Scalability of IDW-CNN The entire IDW is randomly partitioned into three subsets, which contain 10, 10, and 20 thousand images respectively. We evaluate the scalability of IDW-CNN by gradually adding one subset in training. Segmentation accuracies on VOC12 *test* and seg-IDW are reported in Table 2 and 3, respectively. For example, the first model is trained with the first 10 thousand samples and the number of samples is doubled (20 thousand) in the second model. The third model is trained with the full IDW (40 thousand). Table 2 shows that the accuracies increase when we simply double the scale of IDW. For instance, IDW-CNN trained with full IDW achieves the best performance. It outperforms the other two models by 3.4% and 1.1% respectively. Another interesting observation is that performances of nearly all object categories can be improved, when presenting more data of IDW. This may because IDW-CNN learns from more diverse data, increasing its modeling complexity. Similar trend is observed in Table 3, where accuracies have much larger room for improvements compared to VOC12, showing that seg-IDW is a competitive complementary test set to evaluate segmentation methods.

Object Interaction Prediction We study the performance of predicting object interactions on int-IDW. To exhibit the superiority of IDW-CNN, we use the two strong

Table 2: Per-class comparisons on VOC12 *test*. Best result is highlighted.

	areo	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mIoU
ResNet-101	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	74.2
IDW-CNN(10k)	91.4	**68.1**	85.0	71.3	82.3	93.8	87.7	88.8	51.7	81.1	73.8	89.1	80.3	89.8	87.2	71.8	91.3	70.9	90.0	77.1	81.8
IDW-CNN(20k)	94.5	67.3	93.1	69.5	83.0	95.1	89.4	93.2	52.0	94.8	75.5	92.8	95.3	91.6	89.1	73.7	**93.7**	74.9	93.9	80.5	85.2
IDW-CNN(40k)	**94.8**	67.3	93.4	**74.8**	**84.6**	**95.3**	89.6	93.6	**54.1**	**94.9**	**79.0**	**93.3**	**95.5**	**91.7**	**89.2**	**77.5**	**93.7**	**79.2**	**94.0**	**80.8**	**86.3**
IDW-CNN w/o OPS - 1	93.6	62.1	91.3	64.3	75.4	91.9	87.4	90.7	34.4	88.1	69.0	86.5	90.1	85.7	85.8	66.4	89.5	58.6	86.2	71.3	79.2
DeepLab2+CRF [3]	92.6	60.4	91.6	63.4	76.3	95.0	88.4	92.6	32.7	88.5	67.6	89.6	92.1	87.0	87.4	63.3	88.3	60.0	86.8	74.5	79.7
CentraleSupelec [1]	92.9	61.2	91.0	66.3	77.7	**95.3**	88.9	92.4	33.8	88.4	69.1	89.8	92.9	87.7	87.5	62.6	89.9	59.2	87.1	74.2	80.2
LRR-4x [5]	92.4	45.1	**94.6**	65.2	75.8	95.1	89.1	92.3	39.0	85.7	70.4	88.6	89.4	88.6	86.6	65.8	86.2	57.4	85.7	77.3	79.3
HP[32]	91.9	48.1	93.4	69.3	75.5	94.2	87.5	92.8	36.7	86.9	65.2	89.1	90.2	86.5	87.2	64.6	90.1	59.7	85.5	72.7	79.1
DPN[15] [15]	89.0	61.6	87.7	66.8	74.7	91.2	84.3	87.6	36.5	86.3	66.1	84.4	87.8	85.6	85.4	63.6	87.3	61.3	79.4	66.4	77.5
RNN[34]	90.4	55.3	88.7	68.4	69.8	88.3	82.4	85.1	32.6	78.5	64.4	79.6	81.9	86.4	81.8	58.6	82.4	53.5	77.4	70.1	74.7
Piecewise[11]	87.5	37.7	75.8	57.4	72.3	88.4	82.6	80.0	33.4	71.5	55.0	79.3	78.4	81.3	82.7	56.1	79.8	48.6	77.1	66.3	70.7
Zoom-out[20]	85.6	37.3	83.2	62.5	66.0	85.1	80.7	84.9	27.2	73.2	57.5	78.1	79.2	81.1	77.1	53.6	74.0	49.2	71.7	63.3	69.6
FCN[16]	76.8	34.2	68.9	49.4	60.3	75.3	74.7	77.6	21.4	62.5	46.8	71.8	63.9	76.5	73.9	45.2	72.4	37.4	70.9	55.1	62.2
WSSL(weak)+CRF[22]	94.7	62.3	93.3	65.5	75.8	94.6	**89.7**	**93.9**	38.6	93.8	72.2	91.4	95.5	89.0	88.4	66.0	94.5	60.4	91.3	74.1	81.9
BoxSup[4]	89.8	38.0	89.2	68.9	68.0	89.6	83.0	87.7	34.4	83.6	67.1	81.5	83.7	85.2	83.5	58.6	84.9	55.8	81.2	70.7	75.2

Table 3: Per-class comparisons on seg-IDW. Best result is highlighted.

	areo	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mIoU
ResNet-101	50.9	42.0	67.9	17.4	46.4	65.4	59.6	64.8	32.5	21.1	45.8	69.7	74.3	61.2	79.7	25.2	40.0	23.8	34.6	57.6	50.6
IDW-CNN(10k)	61.7	**42.9**	**72.2**	18.4	51.2	66.5	61.3	71.3	35.1	61.6	44.4	74.2	74.6	66.2	79.3	30.9	50.7	24.8	36.0	66.7	55.8
IDW-CNN(20k)	60.5	42.6	70.5	**23.7**	52.0	65.7	61.5	**72.2**	**37.4**	74.1	45.0	74.3	75.2	67.6	80.0	**42.8**	51.3	27.1	**37.5**	65.0	57.6
IDW-CNN(40k)	**64.4**	40.1	**72.2**	21.9	**55.7**	**68.9**	**62.6**	71.7	33.9	**75.6**	51.2	**76.4**	**78.0**	**69.7**	80.1	35.4	**57.6**	**33.7**	**37.5**	**71.6**	**59.1**
IDW-CNN w/o OPS - 1	55.3	37.2	64.8	20.1	54.5	63.5	59.0	67.9	31.8	25.4	**51.5**	71.7	77.1	55.1	80.2	33.5	39.6	32.1	34.9	66.1	52.6
DeepLab2+CRF [3]	50.9	42.0	67.9	17.4	46.4	65.4	59.6	64.8	32.5	21.1	45.8	69.7	74.3	61.2	79.7	25.2	40.0	23.8	34.6	57.6	50.6
WSSL(weak)+CRF[22]	51.4	42.5	61.6	17.0	48.4	62.4	58.3	65.8	34.2	30.8	47.3	70.5	75.1	60.5	**80.4**	34.8	43.6	24.6	33.4	65.9	52.0

Method/Task	Action Pred. Recall-5	Action Pred. Recall-10	Object Pred.
Random Guess	0.0006	0.0012	N/A
IDW-CNN w/o OPS - 1	0.9340	0.9568	0.7954
IDW-CNN w/o OPS - 2	0.9295	0.9591	0.7909
Full Model	**0.9620**	**0.9760**	**0.9523**

Table 4: Results of object interaction prediction.

Method/Task	Action Pred. Recall-5	Action Pred. Recall-10	Object Pred.
Random Guess	0.0006	0.0012	N/A
IDW-CNN w/o OPS - 1	0.0975	0.3048	0.0243
Full Model	**0.5488**	**0.8293**	**0.9512**

Table 5: Results of zero-shot object interaction prediction.

baselines, IDW-CNN w/o OPS-1 and -2. The evaluation metric of object interaction is Recall-n ($n = 5, 10$), measuring the possibility that the true interaction is among the top 5 or 10 predicted interactions. These interactions are ranked according to their confidence scores (which are the responses after softmax function). For example, since we have 22 actions and 20 object categories, the total number of possible configurations of interactions are $20 \times 22 \times 20 = 8800$. Then a random guess results in a Recall-5 of $5 \div 8800 = 0.00057$. Experimental results are shown in Table 4, where IDW-CNN outperforms the others by 3% at Recall-5.

Zero-Shot Prediction Another interesting evaluation is to predict unseen object interaction on zero-IDW, which is not presented in the training stage, such as 'person-ride-cow','dog-suck-bottle', and 'cow-suck-bottle'. Table 5 reports the results. In this setting, IDW-CNN outperforms IDW-CNN w/o OPS-1 with a large margin, *i.e.* 54.88% compared to 9.75% at Recall-5, demonstrating the superior generalization capacity of IDW-CNN. The result of zero-shot interaction prediction is illustrated in the last column of Fig.8. When presenting an image with a man riding a cow, IDW-CNN accurately predict the interaction 'person-ride-cow'. And it also demonstrates advantage in segmenting this image, see the 3rd and 4th row in the last column.

4.2. Segmentation Benchmarks

The segmentation accuracies of IDW-CNN are compared to state-of-the-art methods on both VOC12 *test* and seg-IDW. On VOC12 *test*, we adopt 9 fully-supervised methods, including DeepLab2+CRF [3], CentraleSupelec [1], LRR-4x [5], HP [32], DPN [15], RNN [34], Piecewise [11], Zoom-out [20], and FCN [16]. Two state-of-the art semi-supervised methods are also employed, WSSL(weak)+CRF [22] and BoxSup [4]. Most of these approaches employed pre- and post-processing methods such as multiscale fusion and CRF to improve performance, while IDW-CNN does not.

Results are reported in Table 2. IDW-CNN significantly outperforms the best-performing method by 4.4%. A significant 12% gain is achieved when comparing to ResNet-101, which is the backbone network of IDW-CNN, showing the effectiveness of IDW data and the proposed

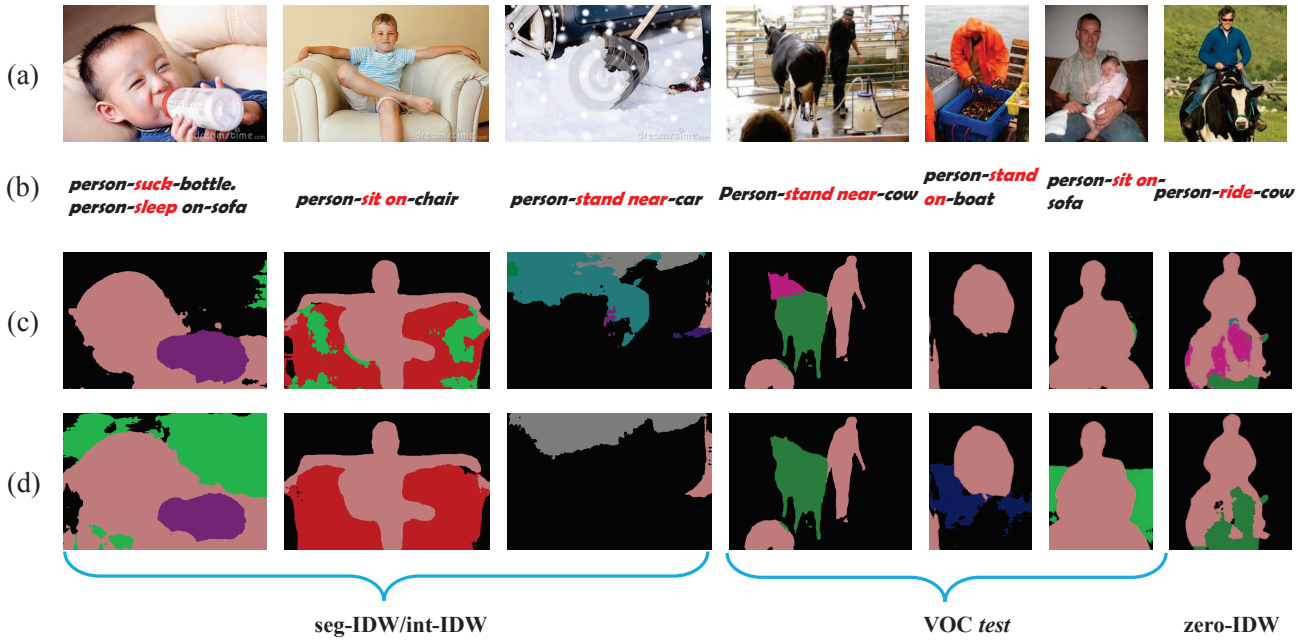


Figure 8: Visualization of object interaction prediction. The first three columns refer to IDW dataset. The middle three columns refer to VOC12 *test* set. The last column refers to zero-shot novel set. (a) are input images. (b) are the object interaction predictions. (c) are segmentation predictions using DeepLab-v2. (d) are segmentation predictions based on our model (IDW-CNN).

network architecture to learn from it. We also compare IDW-CNN with both fully- and semi-supervised methods on seg-IDW. Table 3 shows the results of IDW-CNN and the other competing approaches. IDW-CNN achieves best performances on most of the object categories. Fig.8 visualizes several segmentation and interaction prediction results. Intuitively, IDW-CNN performs very well in both task.

5. Conclusion

We proposed a deep convolutional neural network to increase segmentation accuracy by learning from an Image Descriptions in the Wild (IDW-CNN). IDW-CNN has several appealing properties. First, it fully explores the knowledge from different datasets, thus improves the performance of both dataset. Second, when adding more data to IDW, the segmentation performance in VOC12 can be constantly improved.

IDW-CNN achieves state-of-the-art performance on VOC12, and many valuable facts about semantic image segmentation are revealed through extensive experiments. There are several directions in which we intend to extend this work, such as improving IDW-CNN by adding a knowledge from object attributes. Deeply combining with some language processing techniques also would be a possible way.

Acknowledgement. This work is supported in part by SenseTime Group Limited, the Hong Kong Innovation and Technology Support Programme, and the National Natural Science Foundation of China (61503366, 91320101, 61472410, 61622214). Corresponding authors are Ping Luo and Liang Lin.

References

[1] S. Chandra and I. Kokkinos. Fast, exact and multi-scale inference for semantic image segmentation with deep gaussian crfs. *arXiv preprint arXiv:1603.08358*, 2016. 7

[2] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. *ICLR*, 2014. 2

[3] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *arXiv:1606.00915*, 2016. 2, 4, 5, 7

[4] J. Dai, K. He, and J. Sun. Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. In *ICCV*, pages 1635–1643, 2015. 2, 7

[5] G. Ghiasi and C. C. Fowlkes. Laplacian pyramid reconstruction and refinement for semantic segmentation. In *ECCV*, pages 519–534. Springer, 2016. 7

[6] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CVPR*, 2015. 1

[7] K. He, X. Zhang, S. Ren, and J. Sun. Identity mappings in deep residual networks. *ECCV*, 2016. 1

[8] R. Hu, M. Rohrbach, and T. Darrell. Segmentation from natural language expressions. *ECCV*, 2016. 2

[9] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, M. Bernstein, and L. Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. 2016. 3

[10] D. Lin, J. Dai, J. Jia, K. He, and J. Sun. Scribblesup: Scribble-supervised convolutional networks for semantic segmentation. *CVPR*, 2016. 2

[11] G. Lin, C. Shen, I. Reid, et al. Efficient piecewise training of deep structured models for semantic segmentation. *arXiv preprint arXiv:1504.01013*, 2015. 7

[12] L. Lin, G. Wang, R. Zhang, R. Zhang, X. Liang, and W. Zuo. Deep structured scene parsing by learning with image descriptions. *CVPR*, 2016. 2

[13] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *ECCV*. 1, 5

[14] X. Liu, B. Cheng, S. Yan, J. Tang, T. S. Chua, and H. Jin. Label to region by bi-layer sparsity priors. In *ACM-MM*, pages 115–124. ACM, 2009. 2

[15] Z. Liu, X. Li, P. Luo, C.-C. Loy, and X. Tang. Semantic image segmentation via deep parsing network. In *ICCV*, pages 1377–1385, 2015. 2, 7

[16] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, pages 3431–3440, 2015. 2, 7

[17] C. Lu, R. Krishna, M. Bernstein, and L. Fei-Fei. Visual relationship detection with language priors. In *ECCV*, pages 852–869. Springer, 2016. 6

[18] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, pages 3111–3119, 2013. 3

[19] G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. J. Miller. Introduction to wordnet: An on-line lexical database. *International journal of lexicography*, 3(4):235–244, 1990. 3

[20] M. Mostajabi, P. Yadollahpour, and G. Shakhnarovich. Feedforward semantic segmentation with zoom-out features. In *CVPR*, pages 3376–3385, 2015. 7

[21] W. Ouyang, X. Wang, X. Zeng, S. Qiu, P. Luo, Y. Tian, H. Li, S. Yang, Z. Wang, C.-C. Loy, et al. Deepid-net: Deformable deep convolutional neural networks for object detection. In *CVPR*, pages 2403–2412, 2015. 1

[22] G. Papandreou, L.-C. Chen, K. Murphy, and A. L. Yuille. Weakly-and semi-supervised learning of a dcnn for semantic image segmentation. *ICCV*, 2015. 2, 7

[23] D. Pathak, P. Krahenbuhl, and T. Darrell. Constrained convolutional neural networks for weakly supervised segmentation. In *ICCV*, pages 1796–1804, 2015. 2

[24] D. Pathak, E. Shelhamer, J. Long, and T. Darrell. Fully convolutional multi-class multiple instance learning. *arXiv preprint arXiv:1412.7144*, 2014. 2

[25] P. O. Pinheiro and R. Collobert. From image-level to pixel-level labeling with convolutional networks. In *CVPR*, pages 1713–1721, 2015. 2

[26] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, pages 91–99, 2015. 1

[27] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 115(3):211–252, 2015. 2

[28] A. G. Schwing and R. Urtasun. Fully connected deep structured networks. *arXiv preprint arXiv:1503.02351*, 2015. 2

[29] R. Socher, J. Bauer, C. D. Manning, and A. Y. Ng. Parsing with compositional vector grammars. In *ACL (1)*, pages 455–465, 2013. 3

[30] Y. Sun, Y. Chen, X. Wang, and X. Tang. Deep learning face representation by joint identification-verification. In *NIPS*, pages 1988–1996, 2014. 1

[31] Y. Tian, P. Luo, X. Wang, and X. Tang. Pedestrian detection aided by deep learning semantic tasks. In *CVPR*, pages 5079–5087, 2015. 4

[32] Z. Wu, C. Shen, and A. v. d. Hengel. High-performance semantic segmentation using very deep fully convolutional networks. *arXiv preprint arXiv:1604.04339*, 2016. 7

[33] Z. Zhang, P. Luo, C. C. Loy, and X. Tang. Learning deep representation for face alignment with auxiliary attributes. *PAMI*, 38(5):918–930, 2016. 4

[34] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. Torr. Conditional random fields as recurrent neural networks. In *ICCV*, pages 1529–1537, 2015. 2, 7