Deep-dense Conditional Random Fields for Object Co-segmentation

Zehuan Yuan[1], **Tong Lu**[1*], and **Yirui Wu**[2]
[1]State Key Laboratory for Novel Software Technology, Nanjing University, China
[2]College of Computer and Information, Hohai University, China
zhyuan001@gmail.com, lutong@nju.edu.cn, wuyirui@hhu.edu.cn

Abstract

We address the problem of object co-segmentation in images. Object co-segmentation aims to segment common objects in images and has promising applications in AI agents. We solve it by proposing a co-occurrence map, which measures how likely an image region belongs to an object and also appears in other images. The co-occurrence map of an image is calculated by combining two parts: objectness scores of image regions and similarity evidences from object proposals across images. We introduce a deep-dense conditional random field framework to infer co-occurrence maps. Both similarity metric and objectness measure are learned end-to-end in one single deep network. We evaluate our method on two datasets and achieve competitive performance.

1 Introduction

We consider the task of object co-segmentation. The goal is to segment common objects in a set of images. Specifically, given a set of images, we try to answer two questions: *which objects occur in different images*? and *where are they in each image?* Object co-segmentation has multiple potential applications in AI agents, e.g., we input a query word to any image search engine and it can help summarise a large amount of returned images by picking out common objects inside them. Compared to single foreground segmentation, object co-segmentation can make use of shared information across images to assist final segmentation, which is also the most crucial step for co-segmentation.

Object co-segmentation has been widely researched in recent years [Vicente *et al.*, 2011; Rubinstein *et al.*, 2013; Wang *et al.*, 2013; Zhang *et al.*, 2015; Lee *et al.*, 2015; Quan *et al.*, 2016; Wu *et al.*, 2016]. These methods are mainly different in how to include shared information into image segmentation. For example, [Rubinstein *et al.*, 2013] use SIFT Flow to match pixels across images, [Vicente *et al.*, 2011] match regions, [Zhang *et al.*, 2015] compare object proposals, etc. However, sharing information remains to be a challenge for several reasons. On one hand, low-level matching is noisy and hard to include high-level similarity information. On the other hand, robustly measuring similarity between regions is difficult due to possible illumination changes, viewpoint variations and so on. Another challenge is the diversity of image sets. Other uncommon objects may also appear somewhere. Co-segmentation system must robustly address different scenarios.

In order to overcome these challenges, inspired by [Hayder *et al.*, 2016], we propose an object co-segmentation system that *learns* to share common information for segmentation. Specifically, given a set of images, we first generate multiple object proposals for each of them and every proposal is associated with a label to represent if it is a common object instance or not. All proposals across images are linked in a deep-dense conditional random field (DDCRF), where unary potentials measure how likely an object proposal belongs to a potential cooccurring object and pairwise terms model the compactness between neighbour proposals under different label combinations. The DDCRF is an extension of dense CRF [Krähenbühl and Koltun, 2012] to deep neural network. In a word, its unary and pairwise potential are modelled using deep neural network. By this way, the probability of each proposal to be common corresponds to the marginal distribution of each node and can be easily calculated by performing standard inference on CRF. In addition, we augment DDCRF with a branch to estimate segmentation masks of proposals.

With the common probabilities and segmentation masks of all proposals, we develop a *cooccurrence* map to summerize them for the following segmentation. The *coocurrence* map measures how likely an image pixel belongs to potential common objects and is calculated by accumulating evidences from all proposals. After getting *cooccurrence* maps of all images, object co-segmentation is naturally converted into image foreground segmentation and can be done independently for each image.

In summary, our contributions are twofold: 1) a *cooccurrence* map is proposed to convert object co-segmentation into independent image segmentation, on which tons of systems exist. Every image has a cooccurrence map to encode its shared information across images, which can be used as

*This work is partially supported by the Natural Science Foundation of China under Grant No. 61672273, No. 61272218 and No. 61321491, and the Science Foundation for Distinguished Young Scholars of Jiangsu under Grant No. BK20160021. Correspondence should be addressed to Tong Lu.

extra information for foreground segmentation; 2) a deep-dense conditional random field is introduced to automatically discover common information between object proposals and then calculate *cooccurrence* maps. The entire network is learned in an end-to-end way. To our best knowledge, our system is the first to introduce deep neural network to object co-segmentation. We evaluate our system on two challenging datasets and achieve competitive performance.

2 Related Work

Image co-segmentation aims to segment multiple images in an unified way. The topic has attracted a lot of attentions in recent years because it has proved capable of improving unsupervised foreground segmentation in case that no priors exist for inputing images. Following [Yuan *et al.*, 2014], we roughly categorize co-segmentation into two branches based on how many common classes are considered: object co-segmentation and multi-class image co-segmentation.

Object Co-segmentation. Object co-segmentation aims to segment common objects in multiple images. Most object co-segmentation systems assume that only one single object cooccurs in an input image set. To this end, the popular attempt is to construct saliency maps [Rubinstein *et al.*, 2013; Jerripothula *et al.*, 2016], objectness maps [Vicente *et al.*, 2011] and other feature maps [Mukherjee *et al.*, 2011; Vicente *et al.*, 2011; Wang *et al.*, 2013; Dai *et al.*, 2013] to distinguish foreground objects. Then segmentation masks are propagated across images based on the assumption that matched regions should have similar masks. Images are connected at pixel level [Rubinstein *et al.*, 2013], region level [Rubio *et al.*, 2012; Faktor and Irani, 2013; Lee *et al.*, 2015], object proposal level [Zhang *et al.*, 2015] and even feature-space level [Wang *et al.*, 2013] . Then the problem can be naturally converted into a two-label optimization problem and can be easily solved. Although much better performance is obtained compared to baselines, there still exist a lot of challenges. The key limitation is that manually designed features can not be robust to complex scenarios. [Wang *et al.*, 2013] propose function mapping to link regions but still need a carefully designed features. Recently, [Zhang *et al.*, 2015] first propose to combine Restricted Boltzmann Machines [Bengio, 2009] and Convolutional Neural Network to embed object proposals into a high-dimensional space so that regions can be robustly matched in the space. However, their co-saliency model is still pre-defined instead of automatic learning. [Quan *et al.*, 2016] adopt a manifold ranking algorithm to optimize their constructed superpixel graph. Their method aims to reduce the common strict assumption that cooccurring regions are objects. Similarly, low-level linking is not robust to practical internet images.

Multi-class Object Co-segmentation. Multi-class segmentation systems [Joulin *et al.*, 2012; Kim and Xing, 2012; Ma and Latecki, 2013; Tsai *et al.*, 2016] remove the assumption that one single common object appear in images and aims to segment all common classes. For example, [Kim and Xing, 2012] propose multi-object co-segmentation to segment multiple objects from background using a combinatorial auction optimization framework. [Wu *et al.*, 2016] add human locations as extra priors to segment common foreground objects around a person. [Joulin *et al.*, 2012; Ma and Latecki, 2013; Tsai *et al.*, 2016] even extend it to multi-class image co-segmentation without differentiating objects and background. They iteratively perform discriminative clustering over over-segmented image regions and then multiple class are distinguished from each other. [Yuan *et al.*, 2014] introduce a topic-level random-walk framework to give high votes to cooccurring regions. However, the majority of them still construct across-image relations using predefined features.

3 Our Model

We address the problem of object co-segmentation. To simplify the task and make "common" more clear, we assume that there is one single common object in a given image set. That means other objects will not appear or appear much less. Our system first generates object proposals for every image and our goal is to make use of them to generate the *cooccurrence* map of each image. Deep-dense conditional random field is adopted to encode shared information across images.

3.1 Deep-dense Conditional Random Field

Given a set of n images $\{I_1, I_2, \ldots, I_n\}$, we first generate a pool of object proposals $\mathbf{O} = \{o_1^1, o_2^1, \ldots, o_{g_1}^1, o_1^2, \ldots, o_{g_n}^n\}$ for all images, where g represents the number of proposals in each image. To simplify the denotation, we omit the image superior n and rewrites $\mathbf{O} = \{o_1, o_2, \ldots, o_G\}$. G is the total number of object proposals. Each object proposal o_i has two variables $c_i \in \{0, 1\}$ and m_i. c_i represents if o_i is a common object. $c_i = 0$ means that the proposal belongs to background or an uncommon object. The segmentation mask m_i is to show pixel locations of potential objects in o_i. A deep-dense conditional random field is adopted to model the joint distribution $P(\mathbf{C}, \mathbf{M}|\mathbf{O})$ of $\mathbf{C} = \{c_1, c_2, \ldots, c_G\}$ and $\mathbf{M} = \{m_1, m_2, \ldots, m_G\}$ given $\mathbf{O}$. Formally,

$$P(\mathbf{C}, \mathbf{M}|\mathbf{O}) = \frac{1}{\mathbf{Z}(\mathbf{O})} \exp\Big(-\sum_{i=1}^{G} \phi(c_i, m_i|o_i) - \sum_{i}^{G}\sum_{j>i}^{G} \psi(c_i, c_j, m_i, m_j|o_i, o_j) \Big) \tag{1}$$

where $\phi(\cdot)$ and $\psi(\cdot)$ represent the unary term and pairwise potential, respectively.

Unary Potential. $\phi(c_i, m_i|o_i)$ models how likely the object proposal o_i is a common object and has the segmentation mask m_i. Following the Bayes rule, $\phi(c_i, m_i|o_i)$ can be further decomposed into

$$\phi(c_i, m_i|o_i) = \phi(c_i|o_i) + \phi(m_i|c_i, o_i) \tag{2}$$

However, it is impossible to measure the common characteristic by one single object proposal. Instead, we use objectness v_i as its proxy, which measures the likelihood of o_i as an object. Finally, the final unary potential $\phi(c_i, m_i|o_i)$ is further simplified into $\phi(v_i|o_i) + \phi(m_i|o_i)$ because m_i can be mainly determined by appearance.

Pairwise Potential. To simply formulation, we ignore the influence of masks in pairwise potential. This is reasonable

because a segmentation mask is highly related to appearance [Pinheiro *et al.*, 2015] and does not rely much on predictions from others. Assume that we have extracted feature $\mathbf{f}_i$ for any o_i, then the pairwise potential is reformulated as

$$\psi(c_i, c_j, m_i, m_j | o_i, o_j) = \psi(c_i, c_j | o_i, o_j) = \mu(c_i, c_j)\kappa(\mathbf{f}_i, \mathbf{f}_j) \tag{3}$$

where μ measures the compatibility between labels. We use the typical Potts model $\mu(c_i, c_i) = [\![c_i \neq c_j]\!]$. $\kappa(\cdot, \cdot)$ is a Gaussion kernel

$$\kappa(\mathbf{f}_i, \mathbf{f}_j) = \exp\big(-\frac{1}{2}||\mathbf{f}_i - \mathbf{f}_j||^2\big) \tag{4}$$

Deep Network to Represent Potentials. We use deep convolutional neural network to model unary potentials and extract high-level features of object proposals . VGG16 [Simonyan and Zisserman, 2014] is adopted as our backbone network but the final two fully connected (fc) layers are removed. We add three branches above the final pooling layer (pool5) to represent $\phi(c_i)$, $\phi(m_i)$ and $\mathbf{f}_i$ respectively. The first two branches share fc layers (fc6 and fc7) at the beginning and then divide into two ways to encode $\phi(c_i)$ and $\mathbf{f}_i$ respectively with each following a fc layer. m_i is estimated in another branch by following one convolution layer and two fully connected layers. In order to make outputs having fixed dimension, a $v \times v (v = 40)$ mask is used. In testing stage, the mask will be resized into its original size. In addition, we replace *pool5* with ROI pooling [Ren *et al.*, 2015] to share computations of object proposals in the same image. The detailed architecture is shown in Fig. 1.

3.2 Generate Cooccurrence Maps

For each image I, we use object proposals inside it to generate its *cooccurrence* map R. The element of R represents how confident its corresponding pixel appears in most of other images. In order to generate these maps, we first maximize $P(\mathbf{C}, \mathbf{M}|\mathbf{O})$ and get MAP estimates for object proposals. Note that $P(\mathbf{C}, \mathbf{M}|\mathbf{O})$ can be further decomposed after our simplification for both unary and pairwise potentials

$$P(\mathbf{C}, \mathbf{M}|\mathbf{O}) \propto \exp\Big(\sum_i \big(-\phi(c_i|o_i) - \sum_{j>i} \psi(c_i, c_j|o_i, o_j)\big)\Big) \prod_i \exp(-\phi(m_i|o_i)) \tag{5}$$

Therefore, $\mathbf{M}$ can be inferred independently for each object proposal by only performing forward pass. In term of $\mathbf{C}$, the mean field variational inference [Krähenbühl and Koltun, 2013] is adopted to get MAP estimates. Specifically, $P(\mathbf{C}|\mathbf{O})$ is approximated by $Q(C) = \prod_i \mathbf{q}_i(c_i)$. The marginal distribution $\mathbf{q}_i(c_i)$ of each o_i can be calculated in an iterative way

$$\mathbf{q}_i(c_i) \propto \exp\big(-\phi(c_i) - \sum_{j \neq i} \sum_{c_j} \psi(c_i, c_j)\mathbf{q}_j(c_j)\big) \tag{6}$$

The iteration proceeds until convergence or the iteration step exceeds a threshold. Although the update can be boosted by high-dimensional filtering techniques, we use the standard convolution operator because convolution is fast enough on current GPUs. By this way, we get $\mathbf{q}_i(c_i)$ for any object proposal to encode its probability to appear in other images.

With the marginal distributions and mask predictions of all object proposals inside an image I, it's *cooccurrence* map is calculated by max pooling over them

$$R(x, y) = \max_{i,(x,y)\in i} \mathbf{q}(c_i = 1) m_i(\overline{x}, \overline{y}) \tag{7}$$

where $(\overline{x}, \overline{y})$ corresponds to the relative coordinate of (x, y) in o_i.

Object Co-segmentation. Given cooccurrence maps of all images, object co-segmentation can be converted into simple foreground segmentation for separate images. We adopt dense CRF in [Krähenbühl and Koltun, 2012] as the foreground segmentation method. Specifically, given an image, we use its *cooccurrence* map as the single unary term and a bilateral filter as pairwise potential. By this way, we obtain object segmentations for all images.

4 End-to-end Training

In this section, we will learn the parameters of our deep neural network to make estimated segmentation masks consistent with groundtruth, embedding features capable of comparing proposals with Euclidean metric and finally inferred objects common in training dataset.

4.1 Gradients Computation

Specifically, given the groundtruth labels $(\hat{c_i}, \hat{m_i})$of any o_i, we define the loss function as the summation of two parts

$$\mathcal{L} = \mathcal{L}(\hat{\mathbf{M}}, \mathbf{M}) + \mathcal{L}(\hat{\mathbf{C}}, \mathbf{C}) \tag{8}$$

where $\hat{\mathbf{M}}$ and $\hat{\mathbf{C}}$ correspond to groundtruth mask labels and common objects labels of all proposals, respectively.

In order to define $\mathcal{L}(\hat{\mathbf{M}}, \mathbf{M})$ elegantly, we assume that groundtruth $\hat{m_i} \in \{-1, 1\}^{v \times v}$ with 1 corresponding to object and -1 background. Then $\mathcal{L}(\hat{\mathbf{M}}, \mathbf{M})$ is defined as

$$\mathcal{L}(\hat{\mathbf{M}}, \mathbf{M}) = \sum_i \frac{1 + c_i}{2} \sum_{(x,y)} \hat{w}(x, y) \log\big(1 + e^{-m_i(x,y)\hat{m}_i(x,y)}\big) \tag{9}$$

where the multiplying factor means only segmentation masks of common objects are considered. $\hat{w}(x, y)$ is the weight of the (x, y)-th pixel to balance foreground and background

$$\hat{w}(x, y) = \begin{cases} \frac{1}{2 \times \sum_{(c,d)} [\![\hat{m}(c,d) = 1]\!]} & \hat{m}(x, y) = 1 \\ \frac{1}{2 \times \sum_{(c,d)} [\![\hat{m}(c,d) = -1]\!]} & \hat{m}(x, y) = -1 \end{cases} \tag{10}$$

Note that minimizing $\mathcal{L}(\hat{\mathbf{M}}, \mathbf{M})$ will increase estimated scores for object pixels but reduce them for background. The gradients of all parameters can be calculated easily by back-propagation.

In terms of $\mathcal{L}(\hat{\mathbf{C}}, \mathbf{C})$, we use the standard cross-entropy, namely, $\mathcal{L}(\hat{\mathbf{C}}, \mathbf{C}) = \sum_i -\log(q_i(\hat{c}_i)$. However, the computation of gradients is not trivial due to the pairwise potential. We derive mean-field gradients with respect to network parameters $\boldsymbol{\omega}$ as following

$$\frac{\partial \mathcal{L}(\hat{\mathbf{C}}, \mathbf{C})}{\boldsymbol{\omega}} = \frac{\partial \mathcal{L}(\hat{\mathbf{C}}, \mathbf{C})}{\partial \boldsymbol{\phi}} \frac{\partial \boldsymbol{\phi}^T}{\partial \boldsymbol{\omega}} + \frac{\partial \mathcal{L}(\hat{\mathbf{C}}, \mathbf{C})}{\partial \mathbf{f}} \frac{\partial \mathbf{f}^T}{\partial \boldsymbol{\omega}} \tag{11}$$

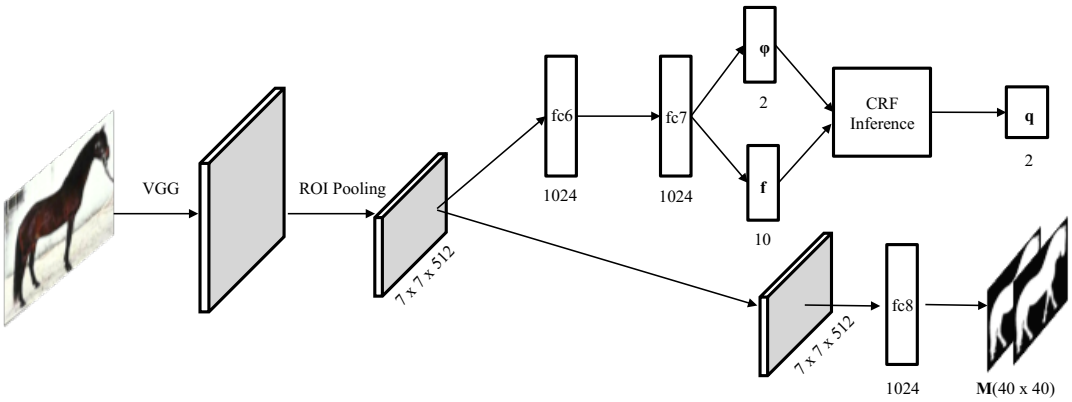

Figure 1: The architecture of DDCRF. The pipeline is based on VGG network. The forward pass goes three ways to model different components of DDCRF.

where $\mathbf{f} = (\mathbf{f}_1^T, \mathbf{f}_1^T, \ldots, \mathbf{f}_G^T)^T$ and $\phi = (\phi_1^T, \phi_1^T, \ldots, \phi_G^T)^T$. Note that $\frac{\partial \phi^T}{\partial \boldsymbol{\omega}}$ and $\frac{\partial \mathbf{f}^T}{\partial \boldsymbol{\omega}}$ can be easily calculated by back-propagation. Now we derive $\frac{\partial L(\hat{\mathbf{C}},\mathbf{C})}{\partial \phi}$ and $\frac{\partial L(\hat{\mathbf{C}},\mathbf{C})}{\partial \mathbf{f}}$.

We rewrite $\mathcal{L}(\hat{\mathbf{C}}, \mathbf{C})$ as $L(\mathbf{q})$ for brevity, where $\mathbf{q}$ is a concatenated vector $(\mathbf{q}_1^T, \mathbf{q}_1^T, \ldots, \mathbf{q}_G^T)^T$ of marginal probabilities. Similarly, let $\boldsymbol{\Phi} = \mathbf{K} \otimes \mu$ where $\mathbf{K}$ is a kernel matrix $[k_{ij}]_{G*G}$with $k_{ij} = \kappa(\mathbf{f}_i, \mathbf{f}_j)$. Following [Krähenbühl and Koltun, 2013], we can compute them iteratively

$$\frac{\mathcal{L}(\mathbf{q})}{\partial \phi} = \sum_{h=1}^{H} \mathbf{b}^{(h)T} \tag{12}$$

$$\frac{\mathcal{L}(\mathbf{q})}{\partial \mathbf{f}_i} = \sum_{h=1}^{H} \sum_{j} \left((\mathbf{f}_j - \mathbf{f}_i) \mathbf{K}_{ij} \mu \mathbf{b}_i^{(h)} \right) \tag{13}$$

where $h = \{1, 2, \ldots, H\}$ represents the iteration step and $\mathbf{q}_j^h$ corresponds to the marginal probabilities in the h step of inference. Simultaneously,

$$\mathbf{b}^{(H)} = A^{(H)} \left(\bigtriangledown \mathcal{L}(\mathbf{q}) \right) \tag{14}$$

$$\mathbf{b}^{(h)} = A^{(h)} \boldsymbol{\Phi} \mathbf{b}^{(h+1)}, h = 1, 2, \ldots, H-1 \tag{15}$$

where $A^{(h)}$ is a block diagonal matrix with each block $A_i^{(h)} = \mathbf{q}_i^{(h)} \mathbf{q}_i^{(h)^T} - \mathrm{diag}(\mathbf{q}_i^h)$. Therefore, both two mean-field gradients can be iteratively refined until convergence. Then backpropagation is performed to update parameters of all layers. The iteration is very fast on GPU because $\frac{\mathcal{L}(\mathbf{q})}{\partial \mathbf{f}}$ can be computed as convolution as shown in Equation 13.

4.2 Training Strategy

The entire network is fine-tuned on a pretrained model for ImageNet classification except new layers. We randomly initialize all new layers by drawing weights from a zero-mean Gaussian distribution with the standard deviation 0.01. In order to share computations among object proposals, we follow the same sampling strategy as [Ren *et al.*, 2015], namely, 2 images are sampled per iteration and 64 object proposals per image. Therefore, positives and negatives account for half respectively in each mini-batch with 128 proposals totally.

In order to make optimization faster and converge to a good optima, a two-stage training is adopted to train the network. Firstly, we learn parameters without including pairwise term. In this case, the training reduces to classify proposals and regress masks similar as in [Pinheiro *et al.*, 2015]. Then we finetune the network by adding pairwise term with customized mini-batches, where a proposal is considered as positive if its overlap with any groundtruth is larger than a threshold (we fixed it 0.7) and its object class is common. In contrast, negatives are composed of a mixture of proposals with maximal overlaps between $[0.1, 0.5]$ and those belonging to other uncommon classes. Uncommon objects are enforced to have less than $num_+/6$ examples so that positives are saliently common. Instead of selecting background regions, we use proposals with the limited overlap range as negatives to make training robust and fast. It has the same effect as hard mining. The two-stage training strategy is necessary because training deep-dense CRF needs stable unary estimation. Direct optimization will mostly cause divergence in our early experiments.

5 Experiments and Discussion

We train DDCRF network on PASCAL VOC 2012 dataset [Everingham *et al.*,] and test the performance on the widely used benchmark iCoseg [Batra *et al.*, 2010] and one more challenging Internet dataset [Rubinstein *et al.*, 2013].

Datasets. We learn parameters on PASCAL VOC 2012 dataset, in which 11540 images have groundtruth detection boxes and 2913 images have segmentation masks. The iCoseg dataset is an object co-segmentation benchmark widely used to evaluate co-segmentation systems, which contains a total of 643 images for 38 object classes. All images are segmented well at pixel level. In contrast, Internet dataset is proposed to evaluate co-segmentation methods on natural images. It contains 3 classes (airplane, car and house) with each class having thousands of downloaded internet images. Compared to iCoseg, images in Internet are mixed with other uncommon objects and multiple background images.

Training Details. We adopt selective search [van de Sande *et al.*, 2011] to generate multiple object proposals in training images. Then mini-batches are sampled from the com-

Figure 2: Segmentation results on iCoseg. Three images (from top to bottom) of three classes (from left to right) are given. The color difference between segmentation masks and original images is for showing saliently.

bination of ground truth bounding boxes and these potentials in the same way as Sec. 4.2. Since there are relatively less images for segmentation masks, joint training will cause severe overfitting. Therefore, we first finetune the network on images with annotated boxes without considering segmentation masks. The training follows two-stage learning strategy as shown in Sec. 4.2 with the initial learning rate 0.001. Stochastic Gradient Descent (SGD) is used as optimizer and the learning rate decreases to 0.0001 after 80K iterations. After finishing the first two-stage training, we add the segmentation branch. In order to prevent overfitting, we fix parameters of the first four convolutional layers and give different learning rates for remaining layers. Specifically, the learning rates of layers in segmentation branch start with 0.001 and others keep training with 0.0001. Additionally, we add dropout layers for all three branches. In the segmentation branch, fc8 is trained with the dropout rate 0.5. Two dropout layers with dropout rate 0.8 are followed after fc6 and fc7, respectively.

Evaluation Metrics. Two standard metrics are used: Precision P and Jaccard index J. P represents the percentage of correctly labeled pixels including both foreground and background, while J focuses on the overlap (Intersection of Union) between estimated object segmentation and foreground.

5.1 Evaluation on ICoseg

iCoseg consists of 38 classes and we evaluate for each class independently. Firstly, selective search [van de Sande *et al.*, 2011] is used to generate a lot of object proposals. Every proposal is forwarded into our network to get its feature $\mathbf{f}$ and unary score ϕ. All proposals are then connected into a dense CRF. Iterative inference is performed to get their marginal probabilities and then further cooccurrence maps. Finally, segmentation masks are obtained separately for each image. We show some segmented examples in Fig. 2. We can see our systems generate promising segmentations for listed classes, which indicates our system is effective to co-segment common objects in this dataset.

Comparison with Existing Systems. Qualitatively, we report both P and J metric in Table. 1. Several comparisons with recent object co-segmentation systems are also included. Besides, we report the performance on sub-iCoseg dataset (Table. 2) to make more comparisons because they only report their performance in this split. sub-iCoseg has 122 images

Table 1: Comparison with existing systems and ablation study on the entire iCoseg

Method	P	J
[Vicente *et al.*, 2011]	85.4	-
[Joulin *et al.*, 2012]	86.3	-
[Rubio *et al.*, 2012]	83.5	-
[Faktor and Irani, 2013]	92.8	0.73
[Lee *et al.*, 2015]	90.5	-
[Quan *et al.*, 2016]	93.3	0.76
[Jerripothula *et al.*, 2016]	93.4	0.77
baseline	88.6	0.78
w/o seg	91.5	0.74
Ours(full)	**94.4**	**0.82**

Table 2: Comparison with existing systems on sub-iCoseg

Method	P	J
[Rubinstein *et al.*, 2013]	89.6	0.68
[Quan *et al.*, 2016]	94.8	0.82
Ours (full)	**96.0**	**0.86**

of 16 classes selected from iCoseg. We get the state-of-the-art performance on both datasets with respect to both metrics . This indicates our deep system can encode shared information effectively and transfer them into segmentation successfully. We analyse that main reasons are twofold: 1) the deep network can learn robust similarity metric compared to other existing systems, which will help co-segmentation a lot with dense CRF; 2) segmentation masks are learned to rectify object proposals so that each object proposal has accurate segmentation mask compared to regard original segments in proposals as final masks.

Ablation Study. We also did ablation study to show the contribution of each component in our full system. Firstly, we remove pairwise potential in DDCRF as our *baseline*. The baseline reduces to simple image segmentation problem without sharing information among images. In *w/o seg*, we do not estimate segmentation masks but use accompanying masks of object proposals to calculate *cooccurrence* maps instead. Dropping each of these components will hurt the final performance. In particular, sharing information among images can help final segmentation of each image, which is consistent with the observations of prior works.

In summary, all scaffoldings of our system are necessary and our full system gives promising performance on iCoseg.

5.2 Evaluation on Internet

Internet images are collected by downloading returned images after querying by key words of three classes, i.e., car, horse and airplane. We use its widely used subset for evaluation, in which each class has 100 images. Similarly, for each class, we generate their object proposals using selective search, then all proposals are connected using dense CRF based on their unary potentials, segmentation masks and embedding features. In Fig. 3, we give some results produced by our full system. Despite large intra-class variations, our

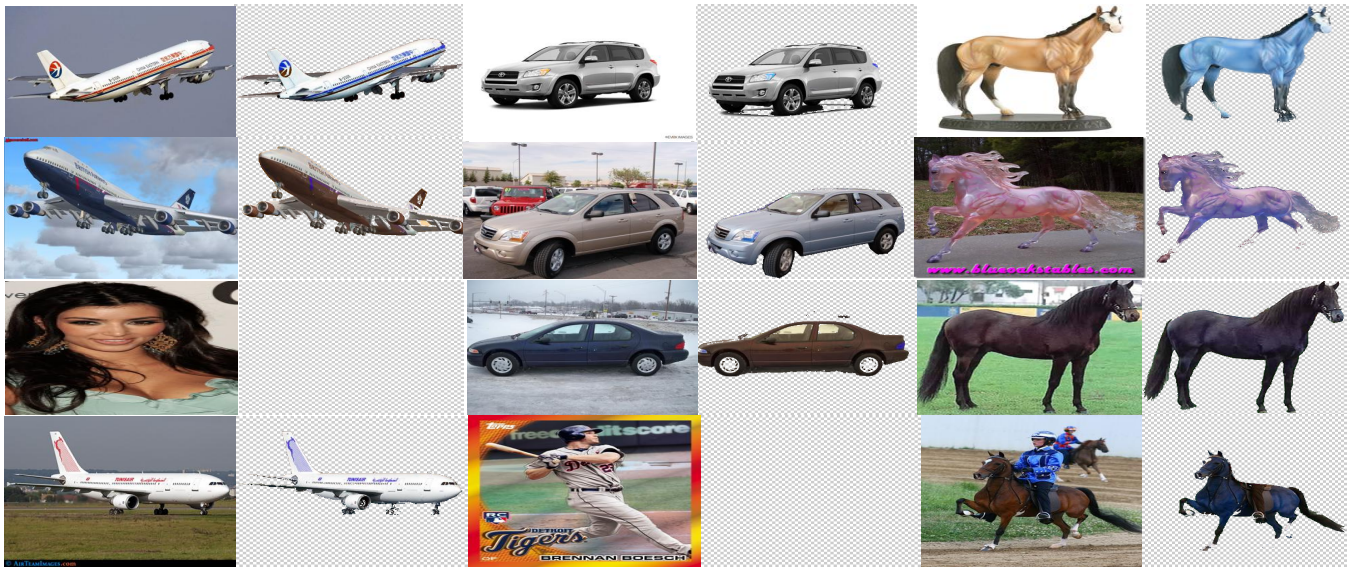

Figure 3: Sampled images and corresponding segmentations of our full system on Internet dataset. For each class, four images are listed. Uncommon objects and background regions are filtered. For saliency, different colors are used for segmented masks. White segmentations mean that these images are background without any common object instances in them. Our full system is robust to different scenarios, e.g., different viewpoints, intra-class variations, background images and uncommon foreground objects.

Table 3: Comparison with state-of-the-art methods and ablation study

Method	Airplane		Car		Horse	
	P	J	P	J	P	J
[Rubinstein *et al.*, 2013]	88.0	0.56	85.4	0.64	82.8	0.52
[Quan *et al.*, 2016]	91.0	0.56	88.5	0.67	89.3	0.58
[Jerripothula *et al.*, 2016]	90.5	0.61	88.0	0.71	88.3	0.61
baseline	89.8	0.59	88.4	0.64	88.2	0.57
w/o seg	88.2	0.56	88.0	0.61	87.8	0.55
Ours	**92.6**	**0.66**	**90.4**	**0.72**	**90.2**	**0.65**

system can obtain good segmentations indicating the learned neural network can help object co-segmentation in images.

Comparison with Existing Systems. Similarly, we compare with several existing methods on both metrics and results are reported in Table. 3. Our full system outperforms all other methods on all three methods, which indicates our method can effectively discover shared information by robustly comparing object proposals. It is worthy noting that the performance increase by a larger margin on Internet compared to iCoseg. The main reason is that each of the classes in Internet has more images and intra-class variations are relatively bigger than iCoseg. Furthermore, there exist several noisy examples including background images and other uncommon objects. Therefore, existing low level systems will be influenced a lot by linking ambiguous regions. Our system has learned robust similarity metric between object proposals and thus robust to complex scenarios.

Ablation Study. In order to evaluate the impact of each component on Internet dataset, we also report the performance of *baseline* and *w/o seg* in Table. 3. The performance is consistent with those on iCoseg, indicating every component is crucial to our system.

6 Discussion and Limitations

Although the proposed method gets promising results on both datasets, there still are some inherent disadvantages. Firstly, our system relies on selective search to generate proposals. If an image is complex enough and no object proposals cover object instances, our system will fail. In this case, the region proposal network can be used as an alternative in future to reduce failures. Secondly, the estimation of segmentation masks is still not accurate enough for complex objects with large occlusion or slim structures. The main reason is that training images with segmentation groundtruth in PASCAL is not enough to generalize very well. Furthermore, the image segmentation method dense-CRF that we used also performs badly for these objects, which will magnify negative effects. Our future work lies on training DDCRF on larger datasets to further improve segmentation quality.

7 Conclusion

In this paper, we propose deep-dense conditional random fields for object co-segmentation in images. A *cooccurrence* map is introduced to summarise shared information after performing inference for constructed DDCRF. Based on *cooccurrence* maps, object co-segmentation is converted into single foreground segmentation, where several excellent systems exist. Our system is trained in an end-to-end way with a two-stage training strategy. We evaluate DDCRF on the benchmark iCoseg and a challenging Internet dataset. Competitive performance on both datasets indicates that our method is effective for object co-segmentation.

References

[Batra *et al.*, 2010] Dhruv Batra, Adarsh Kowdle, Devi Parikh, Jiebo Luo, and Tsuhan Chen. icoseg: Interactive co-segmentation with intelligent scribble guidance. In *The Twenty-Third IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2010, San Francisco, CA, USA, 13-18 June 2010*, pages 3169–3176, 2010.

[Bengio, 2009] Yoshua Bengio. Learning deep architectures for AI. *Foundations and Trends in Machine Learning*, 2(1):1–127, 2009.

[Dai *et al.*, 2013] Jifeng Dai, Ying Nian Wu, Jie Zhou, and Song-Chun Zhu. Cosegmentation and cosketch by unsupervised learning. In *ICCV*, 2013.

[Everingham *et al.*,] Mark Everingham, Luc J. Van Gool, Christopher K. I. Williams, John M. Winn, and Andrew Zisserman. The pascal visual object classes (VOC) challenge 2012.

[Faktor and Irani, 2013] Alon Faktor and Michal Irani. Co-segmentation by composition. In *ICCV*, 2013.

[Hayder *et al.*, 2016] Zeeshan Hayder, Xuming He, and Mathieu Salzmann. Learning to co-generate object proposals with a deep structured network. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 2565–2573, 2016.

[Jerripothula *et al.*, 2016] Koteswar Rao Jerripothula, Jianfei Cai, and Junsong Yuan. Image co-segmentation via saliency co-fusion. *IEEE Trans. Multimedia*, 18(9):1896–1909, 2016.

[Joulin *et al.*, 2012] Armand Joulin, Francis Bach, and Jean Ponce. Multi-class cosegmentation. In *CVPR*, pages 542–549, 2012.

[Kim and Xing, 2012] Gunhee Kim and Eric P. Xing. On multiple foreground cosegmentation. In *CVPR*, pages 837–844, 2012.

[Krähenbühl and Koltun, 2012] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. *CoRR*, abs/1210.5644, 2012.

[Krähenbühl and Koltun, 2013] Philipp Krähenbühl and Vladlen Koltun. Parameter learning and convergent inference for dense random fields. In *ICML*, pages 513–521, 2013.

[Lee *et al.*, 2015] Chulwoo Lee, Won-Dong Jang, Jae-Young Sim, and Chang-Su Kim. Multiple random walkers and their application to image cosegmentation. In *CVPR*, pages 3837–3845, 2015.

[Ma and Latecki, 2013] Tianyang Ma and Longin Jan Latecki. Graph transduction learning with connectivity constraints with application to multiple foreground cosegmentation. In *CVPR*, pages 1955–1962, 2013.

[Mukherjee *et al.*, 2011] Lopamudra Mukherjee, Vikas Singh, and Jiming Peng. Scale invariant cosegmentation for image groups. In *CVPR*, pages 1881–1888, 2011.

[Pinheiro *et al.*, 2015] Pedro H. O. Pinheiro, Ronan Collobert, and Piotr Dollár. Learning to segment object candidates. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 1990–1998, 2015.

[Quan *et al.*, 2016] Rong Quan, Junwei Han, Dingwen Zhang, and Feiping Nie. Object co-segmentation via graph optimized-flexible manifold ranking. In *CVPR*, pages 687–695, 2016.

[Ren *et al.*, 2015] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. In *NIPS*, pages 91–99, 2015.

[Rubinstein *et al.*, 2013] Michael Rubinstein, Armand Joulin, Johannes Kopf, and Ce Liu. Unsupervised joint object discovery and segmentation in internet images. In *CVPR*, pages 1939–1946, 2013.

[Rubio *et al.*, 2012] José C. Rubio, Joan Serrat, Antonio M. López, and Nikos Paragios. Unsupervised co-segmentation through region matching. In *CVPR*, pages 749–756, 2012.

[Simonyan and Zisserman, 2014] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[Tsai *et al.*, 2016] Yi-Hsuan Tsai, Guangyu Zhong, and Ming-Hsuan Yang. Semantic co-segmentation in videos. In *ECCV*, pages 760–775, 2016.

[van de Sande *et al.*, 2011] Koen E. A. van de Sande, Jasper R. R. Uijlings, Theo Gevers, and Arnold W. M. Smeulders. Segmentation as selective search for object recognition. In *ICCV*, pages 1879–1886, 2011.

[Vicente *et al.*, 2011] Sara Vicente, Carsten Rother, and Vladimir Kolmogorov. Object cosegmentation. In *CVPR*, pages 2217–2224, 2011.

[Wang *et al.*, 2013] Fan Wang, Qixing Huang, and Leonidas J. Guibas. Image co-segmentation via consistent functional maps. In *ICCV*, 2013.

[Wu *et al.*, 2016] Chenxia Wu, Jiemi Zhang, Ashutosh Saxena, and Silvio Savarese. Human centred object co-segmentation. *CoRR*, abs/1606.03774, 2016.

[Yuan *et al.*, 2014] Ze-Huan Yuan, Tong Lu, and Palaiahnakote Shivakumara. A novel topic-level random walk framework for scene image co-segmentation. In *ECCV*, pages 695–709, 2014.

[Zhang *et al.*, 2015] Dingwen Zhang, Junwei Han, Chao Li, and Jingdong Wang. Co-saliency detection via looking deep and wide. In *CVPR*, pages 2994–3002, 2015.