UNSUPERVISED IMAGE SEGMENTATION BY BACKPROPAGATION

Asako Kanezaki

National Institute of Advanced Industrial Science and Technology (AIST)
2-4-7 Aomi, Koto-ku, Tokyo 135-0064, Japan

ABSTRACT

We investigate the use of convolutional neural networks (CNNs) for unsupervised image segmentation. As in the case of supervised image segmentation, the proposed CNN assigns labels to pixels that denote the cluster to which the pixel belongs. In the unsupervised scenario, however, no training images or ground truth labels of pixels are given beforehand. Therefore, once when a target image is input, we jointly optimize the pixel labels together with feature representations while their parameters are updated by gradient descent. In the proposed approach, we alternately iterate label prediction and network parameter learning to meet the following criteria: (a) pixels of similar features are desired to be assigned the same label, (b) spatially continuous pixels are desired to be assigned the same label, and (c) the number of unique labels is desired to be large. Although these criteria are incompatible, the proposed approach finds a plausible solution of label assignment that balances well the above criteria, which demonstrates good performance on a benchmark dataset of image segmentation.

Index Terms**—** Convolutional neural networks, Unsupervised learning, Feature clustering

1. INTRODUCTION

Image segmentation has attracted attention in computer vision research for decades. The applications of image segmentation include object detection, texture recognition, and image compression. In the supervised scenario, in which a set of pairs of images and pixel-level semantic labels (such as “sky” or “bicycle”) is used for training, the goal is to train a system that classifies the labels of *known* categories for image pixels. On the other hand, in the unsupervised scenario, image segmentation is used to predict more general labels, such as “foreground” and “background”. The latter case is more challenging than the former, and furthermore, it is extremely hard to segment an image into an arbitrary number (≥ 2) of plausible regions. The present study considers a problem in which an image is partitioned into an arbitrary number of salient or meaningful regions without any previous knowledge.

Once the pixel-level feature representation is obtained, image segments can be obtained by clustering the feature vectors. However, the design of feature representation remains a challenge. The desired feature representation highly depends on the content of the target image. For instance, if the goal is to detect zebras as a foreground, the feature representation should be reactive to black-white vertical stripes. Therefore, the pixel-level features should be descriptive of colors and textures of a local region surrounding each pixel. Recently, convolutional neural networks (CNNs) have been successfully applied to semantic image segmentation (in supervised learning scenarios) for autonomous driving or augmented reality games, for example. CNNs are not often used in fully unsupervised scenarios; however, they have great potential for extracting detailed features from image pixels, which is necessary for unsupervised image segmentation. Motivated by the high feature descriptiveness of CNNs, we present a joint learning approach that predicts, for an arbitrary image input, *unknown* cluster labels and learns optimal CNN parameters for the image pixel clustering. Then, we extract a group of image pixels in each cluster as a segment.

Now, we describe the problem formulation that we solve for image segmentation. Let $\{\boldsymbol{x}_n \in \mathbb{R}^p\}_{n=1}^N$ be a set of p-dimensional feature vectors of image pixels, where N denotes the number of pixels in an input image. We assign cluster labels $\{c_n \in \mathbb{Z}\}_{n=1}^N$ to all of the pixels by $c_n = f(\boldsymbol{x}_n)$, where $f : \mathbb{R}^p \rightarrow \mathbb{Z}$ denotes a mapping function. Here, f can, for instance, be the assignment function that returns the ID of the cluster centroid closest to $\boldsymbol{x}_n$ among k centroids, which are obtained by, *e.g.*, k-means clustering. For the case in which f and the feature representation $\{\boldsymbol{x}_n\}$ are fixed, $\{c_n\}$ are obtained by the above equation. On the other hand, if f and $\{\boldsymbol{x}_n\}$ are trainable, whereas $\{c_n\}$ are given (fixed), then the above equation can be regarded as a standard supervised classification problem. The parameters for f and $\{\boldsymbol{x}_n\}$ in this case can be optimized by gradient descent if f and the feature extraction functions for $\{\boldsymbol{x}_n\}$ are differentiable. However, in the present study, we predict *unknown* $\{c_n\}$ while training the parameters of f and $\{\boldsymbol{x}_n\}$ in a fully unsupervised manner. To put this into practice, we alternatively solve the following two sub-problems: prediction of the optimal $\{c_n\}$ with fixed f and $\{\boldsymbol{x}_n\}$ and training of the parameters of f and $\{\boldsymbol{x}_n\}$ with fixed $\{c_n\}$.

Let us now discuss the characteristics of the cluster labels $\{c_n\}$ necessary for good image segmentation. Similar to previous studies on unsupervised image segmentation [1, 2], we assume that a good image segmentation solution matches well a solution that a human would provide. When a human is asked to segment an image, he/she would most likely create segments, each of which corresponds to the whole or a (salient) part of a single object instance. An object instance tends to contain large regions of similar colors or texture patterns. Therefore, grouping spatially continuous pixels that have similar colors or texture patterns into the same cluster is a reasonable strategy for image segmentation. On the other hand, in order to separate segments from different object instances, it is better to assign different cluster labels to neighboring pixels of dissimilar patterns. To facilitate the cluster separation, we also consider a strategy in which a large number of unique cluster labels is desired. In conclusion, we introduce the following three criteria for the prediction of $\{c_n\}$:

(a) Pixels of similar features are desired to be assigned the same label.

(b) Spatially continuous pixels are desired to be assigned the same label.

(c) The number of unique cluster labels is desired to be large.

Note that these criteria are incompatible so that they are never satisfied perfectly. However, through the gradual optimization that considers all three criteria simultaneously, the proposed system finds a plausible solution of $\{c_n\}$ that balance well these criteria. In Section 2, we describe the proposed iterative approach to predict $\{c_n\}$ that satisfy the above criteria.

2. METHOD

2.1. Constraint on feature similarity

Let us consider the first criterion, which assigns the same label to pixels of similar features. The proposed solution is to apply a linear classifier that classifies the features of each pixel into q classes. In the present paper, we assume the input to be an RGB image $\mathcal{I} = \{\boldsymbol{v}_n \in \mathbb{R}^3\}_{n=1}^N$, where each pixel value is normalized to $[0, 1]$.

We compute a p-dimensional feature map $\{\boldsymbol{x}_n\}$ from $\{\boldsymbol{v}_n\}$ through M convolutional components, each of which consists of a 2D convolution, ReLU activation function, and a batch normalization function, where a batch corresponds to N pixels of a single input image. Here, we set p filters of region size 3×3 for all of the M components. Note that these components for feature extraction are able to be replaced by alternatives such as fully convolutional networks (FCN) [3]. Next, we obtain a response map $\{\boldsymbol{y}_n = W_c\boldsymbol{x}_n + \boldsymbol{b}_c\}_{n=1}^N$ by applying a linear classifier, where $W_c \in \mathbb{R}^{q\times p}$ and $\boldsymbol{b}_c \in \mathbb{R}^q$. We then normalize the response map to $\{\boldsymbol{y}'_n\}$ such that $\{\boldsymbol{y}'_n\}_{n=1}^N$ has zero mean and unit variance. The motivation behind the normalization process is described in Sec. 2.3. Finally, we obtain the cluster label c_n for each pixel by selecting the dimension that has the maximum value in $\boldsymbol{y}'_n$. We herein refer to this classification rule as argmax classification. Intuitively, the above-mentioned processing corresponds to the clustering of feature vectors into q clusters. The ith cluster of the final responses $\{\boldsymbol{y}'_n\}$ can be written as:

$$C_i = \{\boldsymbol{y}'_n \in \mathbb{R}^q \mid y'_{n,i} \geq y'_{n,j}, \ \ \forall j\}, \tag{1}$$

where $y'_{n,i}$ denotes the ith element of $\boldsymbol{y}'_n$. This is equivalent to assigning each pixel to the closest point among the q representative points, which are placed at infinite distance on the respective axis in the q-dimensional space. Note that C_i can be $\emptyset$, and therefore the number of unique cluster labels is arbitrary from 1 to q.

2.2. Constraint on spatial continuity

The basic concept of image pixel clustering is to group similar pixels into clusters (as shown in Sec. 2.1). In image segmentation, however, it is preferable for the clusters of image pixels to be spatially continuous. Here, we add an additional constraint that favors cluster labels that are the same as those of neighboring pixels. We first extract K fine superpixels $\{\mathcal{S}_k\}_{k=1}^K$ (with a large K) from the input image $\mathcal{I} = \{\boldsymbol{v}_n\}_{n=1}^N$, where $\mathcal{S}_k$ denotes a set of the indices of pixels that belong to the kth superpixel. Then, we force all of the pixels in each superpixel to have the same cluster label. More specifically, letting $|c_n|_{n\in\mathcal{S}_k}$ be the number of pixels in $\mathcal{S}_k$ that belong to the c_nth cluster, we select the most frequent cluster label $c_{\max}$, where $|c_{\max}|_{n\in\mathcal{S}_k} \geq |c_n|_{n\in\mathcal{S}_k}$ for all $c_n \in \{1, \ldots, q\}$. The cluster labels are then replaced by $c_{\max}$ for $n \in \mathcal{S}_k$. In the present paper, we use SLIC [4] with $K = 10,000$ for the superpixel extraction.

2.3. Constraint on the number of unique cluster labels

In the unsupervised image segmentation, there is no clue as to how many segments should be generated in an image. Therefore, the number of unique cluster labels should be adaptive to the image content. As described in Sec. 2.1, the proposed strategy is to classify pixels into an arbitrary number $q'(1 \leq q' \leq q)$ of clusters, whereas q is the possibly maximum value of q'. A large q' indicates oversegmentation, whereas a small q' indicates undersegmentation. The aforementioned criteria (a) and (b) only facilitate the grouping of pixels, which could lead to a naive solution that $q' = 1$. To prevent this kind of undersegmentation failure, we introduce the third criterion (c), which is the preference for a large q'.

Our solution is to insert the *intra-axis* normalization process for the response map $\{\boldsymbol{y}_n\}$ before assigning cluster labels via argmax classification. Here, we use batch normalization [5] (where a batch corresponds to N pixels of a single input image), which is described as follows:

$$y'_{n,i} = \frac{y_{n,i} - \mu_i}{\sqrt{\sigma_i^2 + \epsilon}}, \tag{2}$$

where μ_i and σ_i denote the mean and standard deviation of $\{y_{n,i}\}$, respectively. Note that ϵ is a constant that is added to the variance for numerical stability. This operation (also known as whitening) converts the original responses $\{\boldsymbol{y}_n\}$ to $\{\boldsymbol{y}'_n\}$, where each axis has zero mean and unit variance. Then each $y'_{n,i}(i = 1, \ldots, q)$ has an even chance to be the maximum value of $\boldsymbol{y}'_n$ across axes. Even though this operation does not guarantee that every cluster index $i(i = 1, \ldots, q)$

Algorithm 1: Unsupervised image segmentation

```
Input: $\mathcal{I} = \{\boldsymbol{v}_n \in \mathbb{R}^3\}_{n=1}^N$        // RGB image
Output: $\mathcal{L} = \{c_n \in \mathbb{Z}\}_{n=1}^N$        // Label image
$\{W_m, \boldsymbol{b}_m\}_{m=1}^M \leftarrow$ Init()     // Initialize
$\{W_c, \boldsymbol{b}_c\} \leftarrow$ Init()          // Initialize
$\{\mathcal{S}_k\}_{k=1}^K \leftarrow$ GetSuperPixels( $\{\boldsymbol{v}_n\}_{n=1}^N$ )
for $t = 1$ to $T$ do
    $\{\boldsymbol{x}_n\}_{n=1}^N \leftarrow$ GetFeats( $\{\boldsymbol{v}_n\}_{n=1}^N, \{W_m, \boldsymbol{b}_m\}_{m=1}^M$ )
    $\{\boldsymbol{y}_n\}_{n=1}^N \leftarrow \{ W_c \boldsymbol{x}_n + \boldsymbol{b}_c \}_{n=1}^N$
    $\{\boldsymbol{y}'_n\}_{n=1}^N \leftarrow$ Norm( $\{\boldsymbol{y}_n\}_{n=1}^N$ ) // Batch norm.
    $\{c_n\}_{n=1}^N \leftarrow \{ \arg\max \boldsymbol{y}'_n \}_{n=1}^N$  // Assign labels
    for $k = 1$ to $K$ do
        $c_{\max} \leftarrow \arg\max |c_n|_{n \in \mathcal{S}_k}$
        $c'_n \leftarrow c_{\max}$ for $n \in \mathcal{S}_k$
    $\mathscr{L} \leftarrow$ SoftmaxLoss( $\{\boldsymbol{y}'_n, c'_n\}_{n=1}^N$ )
    $\{W_m, \boldsymbol{b}_m\}_{m=1}^M, \{W_c, \boldsymbol{b}_c\} \leftarrow$ Update( $\mathscr{L}$ )
```

achieves the maximum value for any $n (n = 1, \ldots, N)$, because of this operation, many cluster indices will achieve the maximum value for any $n (n = 1, \ldots, N)$. As a consequence, this intra-axis normalization process gives the proposed system a preference for a large q'.

2.4. Learning network by backpropagation

In this section, we describe how to self-train the network for unsupervised image segmentation. Once a target image is input, we alternatively solve the following two sub-problems: prediction of cluster labels with fixed network parameters and training of network parameters with the (fixed) predicted cluster labels. The former corresponds to the forward process of a network followed by the superpixel refinement described in Sec. 2.2. The latter corresponds to the backward process of a network based on gradient descent. As with the case of supervised learning, we calculate the softmax loss (i.e., the cross-entropy loss) between the network responses $\{\boldsymbol{y}'_n\}$ and the refined cluster labels $\{c'_n\}$. Then, we backpropagate the error signals to update the parameters of convolutional filters $\{W_m, \boldsymbol{b}_m\}_{m=1}^M$ as well as the parameters of the classifier $\{W_c, \boldsymbol{b}_c\}$. In the present paper, we use stochastic gradient descent with momentum for updating the parameters. The parameters are initialized with Xavier initialization [6], which samples values from the uniform distribution normalized according to the input and output layer size. We iterate this forward-backward process T times to obtain the final prediction of cluster labels $\{c_n\}$. Algorithm 1 shows the pseudocode for the proposed unsupervised image segmentation algorithm. Figure 1 illustrates the proposed algorithm for training the proposed CNN network.

As shown in Fig. 1, the proposed CNN network is composed of basic functions. The most characteristic part of the proposed CNN is the existence of the batch normalization layer between the final convolution layer and the argmax classification layer. Unlike the supervised learning scenario, in which the target labels are fixed, the batch normalization of responses over axes is necessary for obtaining reasonable labels $\{c_n\}$ (see Sec. 2.3). Moreover, in contrast to supervised learning, there are multiple solutions of $\{c_n\}$ with different network parameters that achieve near zero loss. The value of the learning rate takes control over the balance between parameter updates and clustering, which leads to different solutions of $\{c_n\}$. We empirically found that setting the learning rate to 0.1 (with momentum 0.9) yielded the best results.

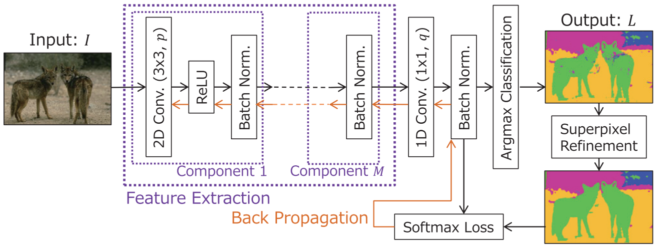


Fig. 1. Illustration of the proposed algorithm for training the proposed CNN network.

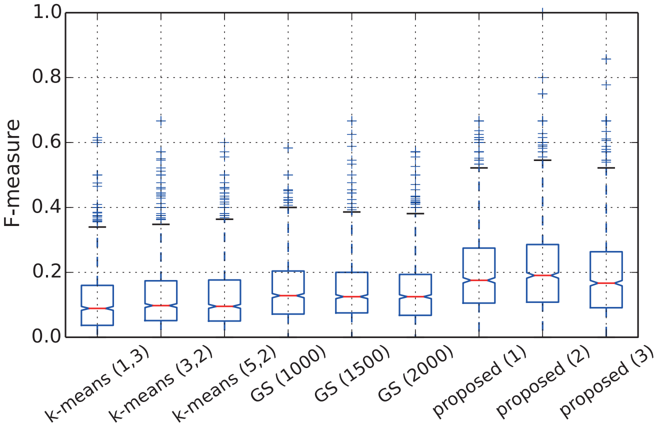


Fig. 3. F-measure per image for various methods.

3. RESULTS

We evaluated the proposed method using 200 test images from the Berkeley Segmentation Dataset and Benchmark (BSDS500) [7, 8]. The test images in this dataset are provided with more than 1,000 hand-labeled segmentations. We trained the proposed CNN model with $T = 500$ iterations for each image, altering the number of convolutional components M as $1, 2, \ldots, 5$. We fixed $p = q = 100$ for all of the experiments. For comparison, we chose to use k-means clustering and the graph-based segmentation method (GS) [9]. For the k-means clustering, we used the concatenation of RGB values in a $\alpha \times \alpha$ window for each pixel representation, where $\alpha = 1, 3, 5, 7$. We extracted connected components as segments from each cluster generated by k-means clustering and the proposed method.

Figure 3 shows the F-measure per image, which is the har-

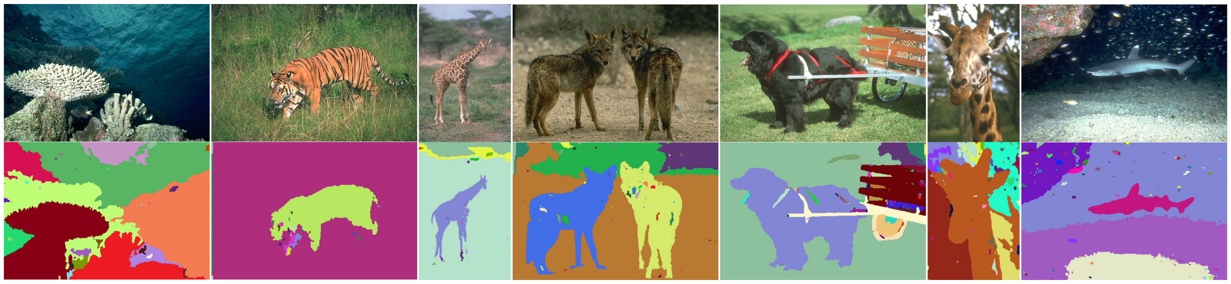
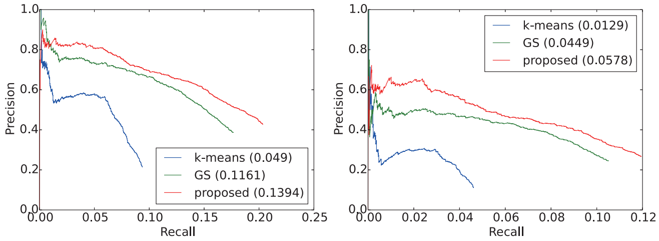


Fig. 2. Example results of the proposed method. Different segments are shown in different colors.



(a) IoU = 0.5

(b) IoU = 0.7

Fig. 4. Precision-recall curves.

monic mean of precision and recall, for various methods. We calculated the intersection over union (IoU) of each estimated segment and ground truth segments and regarded the IoU as correct if the maximum IoU is larger than 0.5. The numbers following the method names in Fig. 3 represent (α, k) for k-means clustering, the similarity threshold β for merging neighboring segments with GS, and M for the proposed method. We chose the best results from $k = 2, 3, \ldots, 20$ for k-means clustering and β = 100, 500, 1,000, 1,500, 2,000 for GS, which are all outperformed by the proposed method. We also show the precision-recall curves with an IoU threshold 0.5 and 0.7 in Fig. 4. Here, segments were arranged in order of decreasing size. The best average precision scores for each method with IoU $= (0.5, 0.7)$ are $(0.049, 0.0129)$ with k-means clustering, $(0.1161, 0.0449)$ with GS, and $(0.1394, 0.0578)$ with the proposed method, which demonstrates the effectiveness of the proposed method. Figure 2 shows typical example results obtained using the proposed method. Many meaningful segments with various colors and textures (such as a tiger and a giraffe) are successfully detected by the proposed method.

4. RELATED WORK

Semantic image segmentation based on CNN has been gaining attention in the literature [10, 11, 3, 12]. Existing work often uses object detectors [13, 14, 15] or user inputs [16, 17] to determine parameters for segmentation. Since pixel-level annotations for image segmentation are difficult to obtain, weakly supervised learning approaches using object bounding boxes [18, 19] or image-level class labels [20, 21, 22, 23] for training are widely used. However, to the best of our knowledge, no studies have considered CNN for image segmentation in a fully unsupervised manner.

Unsupervised deep learning approaches have focused mainly on learning high-level feature representations using generative models [24, 25, 26]. The motivation behind these studies is closely related to the conjecture in neuroscience that there exist neurons that represent specific semantic concepts. Here, we are more interested in the application of deep learning to image segmentation, and thus emphasize the importance of high-level features extracted with convolutional layers. Deep CNN filters are known to be effective for texture recognition and segmentation [27, 28].

Note that the convolution filters used in the proposed method are *trainable* in the standard backpropagation algorithm, although there are no ground truth labels. The present study is therefore related to recent research on deep embedded clustering (DEC) [29]. The DEC algorithm iteratively refines clusters by minimizing the KL divergence loss between soft-assigned data points with an auxiliary target distribution, whereas the proposed method simply minimizes the softmax loss based on the estimated clusters. Similar approaches, such as maximum margin clustering [30] and discriminative clustering [31, 32], have been proposed for semi-supervised learning frameworks, whereas the proposed method is focused on the fully unsupervised image segmentation task.

5. CONCLUSION

We presented a novel CNN architecture and its self-training process that enables image segmentation in an unsupervised manner. Using the backpropagation of the softmax loss to the normalized responses of convolutional layers, the proposed CNN jointly assigned cluster labels to image pixels and updated the convolutional filters to achieve better separation of clusters. We also introduced a superpixel refinement process to achieve the spatial continuity constraint for the estimated segments. Experimental results on the BSDS500 benchmark dataset demonstrated the effectiveness of the proposed method. An interesting direction for future research is to investigate constraints other than superpixel refinement, *e.g.*, using edge density for the segmentation priors.

ACKNOWLEDGMENT

The author is thankful to Dr. Masayuki Tanaka for his valuable comment.

6. REFERENCES

[1] Ranjith Unnikrishnan, Caroline Pantofaru, and Martial Hebert, "Toward objective evaluation of image segmentation algorithms," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, 2007.

[2] Allen Y. Yang, John Wright, Yi Ma, and Shankar Sastry, "Unsupervised segmentation of natural images via lossy data compression," *Computer Vision and Image Understanding*, vol. 110, no. 2, 2008.

[3] Jonathan Long, Evan Shelhamer, and Trevor Darrell, "Fully convolutional networks for semantic segmentation," in *CVPR*, 2015.

[4] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk, "Slic superpixels compared to state-of-the-art superpixel methods," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, 2012.

[5] Sergey Ioffe and Christian Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *ICML*, 2015.

[6] Xavier Glorot and Yoshua Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *AISTATS*, 2010.

[7] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik, "Contour detection and hierarchical image segmentation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 33, no. 5, 2011.

[8] David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *ICCV*, 2001.

[9] Pedro F. Felzenszwalb and Daniel P. Huttenlocher, "Efficient graph-based image segmentation," *Int. J. of Computer Vision*, vol. 59, no. 2, 2004.

[10] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2017.

[11] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected crfs," in *ICLR*, 2015.

[12] Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip H. S. Torr, "Conditional random fields as recurrent neural networks," in *ICCV*, 2015.

[13] Joseph Tighe and Svetlana Lazebnik, "Finding things: Image parsing with regions and per-exemplar detectors," in *CVPR*, 2013.

[14] Bharath Hariharan, Pablo Arbeláez, Ross Girshick, and Jitendra Malik, "Simultaneous detection and segmentation," in *ECCV*, 2014.

[15] Jifeng Dai, Kaiming He, and Jian Sun, "Instance-aware semantic segmentation via multi-task network cascades," in *CVPR*, 2016.

[16] Wenxian Yang, Jianfei Cai, Jianmin Zheng, and Jiebo Luo, "User-friendly interactive image segmentation through unified combinatorial user inputs," *IEEE Transactions on Image Processing*, vol. 19, no. 9, 2010.

[17] Di Lin, Jifeng Dai, Jiaya Jia, Kaiming He, and Jian Sun, "Scribblesup: Scribble-supervised convolutional networks for semantic segmentation," in *CVPR*, 2016.

[18] Jun Zhu, Junhua Mao, and Alan L. Yuille, "Learning from weakly supervised data by the expectation loss svm (e-svm) algorithm," in *NIPS*, 2014.

[19] Feng-Ju Chang, Yen-Yu Lin, and Kuang-Jui Hsu, "Multiple structured-instance learning for semantic segmentation with uncertain training data," in *CVPR*, 2014.

[20] Deepak Pathak, Philipp Krahenbuhl, and Trevor Darrell, "Constrained convolutional neural networks for weakly supervised segmentation," in *ICCV*, 2015.

[21] Niloufar Pourian, Sreejith Karthikeyan, and Bangalore S. Manjunath, "Weakly supervised graph based semantic segmentation by learning communities of image-parts," in *ICCV*, 2015.

[22] Zhiyuan Shi, Yongxin Yang, Timothy Hospedales, and Tao Xiang, "Weakly-supervised image annotation and segmentation with objects and attributes," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2016.

[23] Wataru Shimoda and Keiji Yanai, "Distinct class-specific saliency maps for weakly supervised semantic segmentation," in *ECCV*, 2016.

[24] Honglak Lee, Peter Pham, Yan Largman, and Andrew Y. Ng, "Unsupervised feature learning for audio classification using convolutional deep belief networks," in *NIPS*, 2009.

[25] Quoc V. Le, "Building high-level features using large scale unsupervised learning," in *ICASSP*, 2013.

[26] Honglak Lee, Roger Grosse, Rajesh Ranganath, and Andrew Y. Ng, "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations," in *ICML*, 2009.

[27] Mircea Cimpoi, Subhransu Maji, and Andrea Vedaldi, "Deep filter banks for texture recognition and segmentation," in *CVPR*, 2015.

[28] Bharath Hariharan, Pablo Arbeláez, Ross Girshick, and Jitendra Malik, "Hypercolumns for object segmentation and fine-grained localization," in *CVPR*, 2015.

[29] Junyuan Xie, Ross Girshick, and Ali Farhadi, "Unsupervised deep embedding for clustering analysis," in *ICML*, 2016.

[30] Linli Xu, James Neufeld, Bryce Larson, and Dale Schuurmans, "Maximum margin clustering," in *NIPS*, 2005.

[31] Francis R. Bach and Zaïd Harchaoui, "Diffrac: a discriminative and flexible framework for clustering," in *NIPS*, 2008.

[32] Armand Joulin, Francis Bach, and Jean Ponce, "Discriminative clustering for image co-segmentation," in *CVPR*, 2010.