Unsupervised Learning of Image Segmentation Based on Differentiable Feature Clustering

Wonjik Kim, *Member, IEEE*, Asako Kanezaki, *Member, IEEE*, and Masayuki Tanaka, *Member, IEEE*

***Abstract*— The usage of convolutional neural networks (CNNs) for unsupervised image segmentation was investigated in this study. Similar to supervised image segmentation, the proposed CNN assigns labels to pixels that denote the cluster to which the pixel belongs. In unsupervised image segmentation, however, no training images or ground truth labels of pixels are specified beforehand. Therefore, once a target image is input, the pixel labels and feature representations are jointly optimized, and their parameters are updated by the gradient descent. In the proposed approach, label prediction and network parameter learning are alternately iterated to meet the following criteria: (a) pixels of similar features should be assigned the same label, (b) spatially continuous pixels should be assigned the same label, and (c) the number of unique labels should be large. Although these criteria are incompatible, the proposed approach minimizes the combination of similarity loss and spatial continuity loss to find a plausible solution of label assignment that balances the aforementioned criteria well. The contributions of this study are four-fold. First, we propose a novel end-to-end network of unsupervised image segmentation that consists of normalization and an argmax function for differentiable clustering. Second, we introduce a spatial continuity loss function that mitigates the limitations of fixed segment boundaries possessed by previous work. Third, we present an extension of the proposed method for segmentation with scribbles as user input, which showed better accuracy than existing methods while maintaining efficiency. Finally, we introduce another extension of the proposed method: unseen image segmentation by using networks pre-trained with a few reference images without re-training the networks. The effectiveness of the proposed approach was examined on several benchmark datasets of image segmentation.**

***Index Terms*— Convolutional neural networks, unsupervised learning, feature clustering.**

I. Introduction

IMAGE segmentation has garnered attention in computer vision research for decades. The applications of image segmentation include object detection, texture recognition, and image compression. In supervised image segmentation, a set consisting of pairs of images and pixel-level semantic labels, such as "sky" or "bicycle", is used for training. The objective is to train a system that classifies the labels of the *known* categories for the image pixels. In contrast, unsupervised image segmentation is used to predict more general labels, such as "foreground" and "background". The latter is more challenging than the former. Furthermore, it is extremely difficult to segment an image into an arbitrary number (≥ 2) of plausible regions. This 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 depends considerably on the content of the target image. For instance, if the objective 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 the 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 such as autonomous driving and augmented reality games. CNNs are not often used in completely unsupervised scenarios; however, they have great potential for extracting detailed features from image pixels, which is necessary for unsupervised image segmentation. Driven by the high feature descriptiveness of CNNs, a joint learning approach is presented that predicts, for an arbitrary image input, *unknown* cluster labels and learns the optimal CNN parameters for the image pixel clustering. Subsequently, a group of image pixels in each cluster as a segment is extracted.

The characteristics of the cluster labels that are necessary for good image segmentation are discussed further. Similar to previous studies on unsupervised image segmentation [1], [2], it is assumed that a good image segmentation solution matches well with a solution that a human would provide. When a human is asked to segment an image, they 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. To separate segments from different object instances, it is better to assign different cluster labels to the neighboring pixels of dissimilar patterns. To facilitate the cluster separation, a strategy in which a large number of unique cluster labels is desired is considered

Manuscript received February 14, 2020; revised June 25, 2020; accepted July 17, 2020. Date of current version July 31, 2020. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Jiwen Lu. *(Wonjik Kim and Asako Kanezaki contributed equally to this work.) (Corresponding author: Wonjik Kim.)*

The authors are with the Tokyo Institute of Technology, Tokyo 152-8550, Japan, and also with the National Institute of Advanced Industrial Science and Technology, Tokyo 135-0064, Japan (e-mail: wkim@ok.sc.e.titech.ac.jp; kanezaki@c.titech.ac.jp; mtanaka@sc.e.titech.ac.jp).

Digital Object Identifier 10.1109/TIP.2020.3011269

as well. In conclusion, the following three criteria for the prediction of cluster labels are introduced:

(a) Pixels of similar features should be assigned the same label.
(b) Spatially continuous pixels should be assigned the same label.
(c) The number of unique cluster labels should be large.

In this paper, we propose a CNN-based algorithm that jointly optimizes feature extraction functions and clustering functions to satisfy these criteria. Here, in order to enable end-to-end learning of a CNN, an iterative approach to predict cluster labels using differentiable functions is proposed. The code is available online.[1]

This study is an extension of the previous research published in the international conference on acoustics, speech and signal processing (ICASSP) 2018 [3]. In the previous work, superpixel extraction using simple linear iterative clustering [4] was employed for criterion (b). However, the previous algorithm had a limitation that the boundaries of the segments were fixed in the superpixel extraction process. In this study, a spatial continuity loss is proposed as an alternative to mitigate the aforementioned limitation. In addition, two new applications based on our improved unsupervised segmentation method are introduced: segmentation with user input and utilization of network weights obtained using unsupervised learning of different images. As the proposed method is completely unsupervised, it segments images based on their nature, which is not always related with the user's intention. As an exemplar application of the proposed method, scribbles were used as user input and the effect was compared with other existing methods. Subsequently, the proposed method incurred a high calculation cost to iteratively obtain the segmentation results of a single input image. Therefore, as another potential application of the proposed method, the network weights pre-trained with several reference images were used. Once the network weights are obtained from several images using the proposed algorithm, a new unseen image can be segmented by the fixed network, provided it is somewhat similar to the reference images. The utilization of this technique for a video segmentation task was demonstrated as well.

The contributions of this paper are summarized as follows.

- We proposed a novel end-to-end differentiable network of unsupervised image segmentation.
- We introduced a spatial continuity loss function that mitigated the limitations of our previous method [3].
- We presented an extension of the proposed method for segmentation with scribbles as user input, which showed better accuracy than existing methods while maintaining efficiency.
- We introduced another extension of the proposed method: unseen image segmentation by using networks pre-trained with a few reference images without re-training the networks.

[1] https://github.com/kanezaki/pytorch-unsupervised-segmentation-tip/

II. Related Work

Image segmentation is the process of assigning labels to all the pixels within an image such that the pixels sharing certain characteristics are assigned the same labels. Classical image segmentation can be performed by *e.g.* k-means clustering [5], which is a de facto standard method for vector quantization. The k-means clustering aims to assign the target data to k clusters in which each datum belongs to the cluster with the nearest mean. The graph-based segmentation method (GS) [6] is another example that makes simple greedy decisions of image segmentation. It produces segmentation results that follow the global features of not being too coarse or too fine based on a particular region comparison function. Similar to the classical methods, the proposed method in this study aims to perform unsupervised image segmentation. In recent, there have been proposed a few methods on learning based unsupervised image segmentation [7]–[9]. MsLRR [7] is an efficient and versatile approach that can be switched to both unsupervised and supervised methods. MsLRR [7] employed superpixels (as our previous work [3]), which caused a limitation that the boundaries are fixed to those of the superpixels. W-Net [8] performs unsupervised segmentation by estimating segmentation from an input image and restoring the input image from the estimated segmentation. Therefore, similar pixels are assigned to the same label, though it does not estimate the boundary of each segment. Croitoru *et al.* [9] proposed an unsupervised segmentation method based on deep neural network techniques. Whereas this method performs binary foreground/background segmentation, our method generates arbitrary number of segments. A comprehensive survey about deep learning techniques for image segmentation is presented in [10].

The remainder of this section introduces image segmentation with user input, weakly-supervised image segmentation based on CNN, and methods for unsupervised deep learning.

A. Image Segmentation With User Input

Graph cut is a common method for image segmentation that works by minimizing the cost of a graph where image pixels correspond to the nodes. This algorithm can be applied to image segmentation with certain user inputs such as scribbles [11] and bounding boxes [12]. Image matting is commonly used for image segmentation with user input [13], [14] as well. The distinguishing characteristic of image matting is the soft assignment of pixel labels, whereas, graph cuts produce hard segmentation where every pixel belongs to either the foreground or background. Constrained random walks [15] is proposed to achieve interactive image segmentation with a more flexible user input, which allows scribbles to specify the boundary regions as well as the foreground/background seeds. Recently, a quadratic optimization problem related to dominant-set clusters has been solved with several types of user input: scribbles, sloppy contours, and bounding boxes [16].

The abovementioned methods chiefly produce a binary map that separates image pixels into foreground and background. In order to apply the graph cut to multi-label segmentation

problems, the α-β swap and α-expansion algorithms were proposed in [17]. Both algorithms process repeatedly to find the global minimum of a binary labeling problem. In α-expansion algorithm, an expansion move is defined for a label α to increase the set of pixels that are given this label. This algorithm finds a local minimum such that no expansion move for any label α yields a labeling with lower energy. A swap move takes some subset of the pixels presently labeled α and assigns them the label β and vice versa for a pair of labels α, β. The α-β swap algorithm finds a minimum state such that there is no swap move for any pair of labels α, β that produces a lower energy labeling.

B. Weakly-Supervised Image Segmentation Based on CNN

Semantic image segmentation based on CNNs have been gaining importance in the literature [18]–[21]. As pixel-level annotations for image segmentation are difficult to obtain, weakly supervised learning approaches using object detectors [22]–[24], object bounding boxes [25], [26], image-level class labels [27]–[30], or scribbles [31]–[33] for training are widely used.

Most of the weakly supervised segmentation algorithms [25], [26], [30], [31] generate a *training target* from the weak labels and update their models using the generated training set. Therefore, these methods follow an iterative process that alternates between two steps: (1) gradient descent for training a CNN-based model from the generated target and (2) training target generation by the weak labels. For example, ScribbleSup [31] propagates the semantic labels of scribbles to other pixels using super-pixels so as to completely annotate the images (step 1) and learns a convolutional neural network for semantic segmentation with the annotated images (step 2). In the case of e-SVM [25], the segment proposals from bounding box annotations or pixel level annotations using CPMC segments [34] (step 1) are generated and the model is trained with the generated segment proposals (step 2). Shimoda and Yanai [30] estimated class saliency maps using image level annotation (step 1) and applied fully-connected CRF [35] with the estimated saliency maps as unary potential (step 2). These iterative processes are exposed to danger that the convergence is not guaranteed. The error in training target generation with weak labels might reinforce the entire algorithm to update the model in an undesired direction. Therefore, recent approaches [32], [33], [36] for avoiding the error in training target generation with weak labels are proposed. In this study, to deal with the convergence problem, an *end-to-end* differentiable segmentation algorithm based on a CNN is proposed.

C. Unsupervised Deep Learning

Unsupervised deep learning approaches are mainly focused on learning high-level feature representations using generative models [37]–[39]. The idea behind these studies is closely related to the conjecture in neuroscience that there exist neurons that represent specific semantic concepts. In contrast, the application of deep learning to image segmentation and importance of high-level features extracted with convolutional layers are investigated in this study. Deep CNN filters are known to be effective for texture recognition and segmentation [40], [41].

Notably, 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 the recent research on deep embedded clustering (DEC) [42]. The DEC algorithm iteratively refines clusters by minimizing the KL divergence loss among the 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 [43] and discriminative clustering [44], [45] have been proposed for semi-supervised learning frameworks; however, the proposed method is focused on completely unsupervised image segmentation.

III. METHOD

The problem that is solved for image segmentation is described as follows. For simplicity, let $\{\}$ denote $\{\}_{n=1}^{N}$ unless otherwise noted, where N denotes the number of pixels in an input color image $\mathcal{I} = \{\boldsymbol{v}_n \in \mathbb{R}^3\}$. Let $f : \mathbb{R}^3 \to \mathbb{R}^p$ be a feature extraction function and $\{\boldsymbol{x}_n \in \mathbb{R}^p\}$ be a set of p-dimensional feature vectors of image pixels. Cluster labels $\{c_n \in \mathbb{Z}\}$ are assigned to all of the pixels by $c_n = g(\boldsymbol{x}_n)$, where $g : \mathbb{R}^p \to \mathbb{Z}$ denotes a mapping function. Here, g can be an assignment function that returns the label of the cluster centroid closest to $\boldsymbol{x}_n$. For the case in which f and g are fixed, $\{c_n\}$ are obtained using the abovementioned equation. Conversely, if f and g are trainable whereas $\{c_n\}$ are specified (fixed), then the aforementioned equation can be regarded as a standard supervised classification problem. The parameters for f and g in this case can be optimized by gradient descent if f and g are differentiable. However, in the present study, *unknown* $\{c_n\}$ are predicted while training the parameters of f and g in a completely unsupervised manner. To put this into practice, the following two sub-problems were solved: prediction of the optimal $\{c_n\}$ with fixed f and g and training of the parameters of f and g with fixed $\{c_n\}$.

Notably, the three criteria introduced in Sec. I are incompatible and are never satisfied perfectly. One possible solution for addressing this problem using a classical method is: applying k-means clustering to $\{\boldsymbol{x}_n\}$ for (a), performing graph cut algorithm [17] using distances to centroids for (b), and determining k in k-means clustering using a non-parametric method for (c). However, these classical methods are only applicable to *fixed* $\{\boldsymbol{x}_n\}$ and therefore the solution can be suboptimal. Therefore, a CNN-based algorithm is proposed to solve the problem. The feature extraction functions for $\{\boldsymbol{x}_n\}$ and $\{c_n\}$ are jointly optimized in a manner that satisfies all the aforementioned criteria. In order to enable end-to-end learning of a CNN, an iterative approach to predict $\{c_n\}$ using differentiable functions is proposed.

A CNN structure is proposed, as shown in Fig. 1, along with a loss function to satisfy the three criteria described in Sec. I. The concept of the proposed CNN architecture for considering criteria (a) and (c) is detailed in Section III-A.

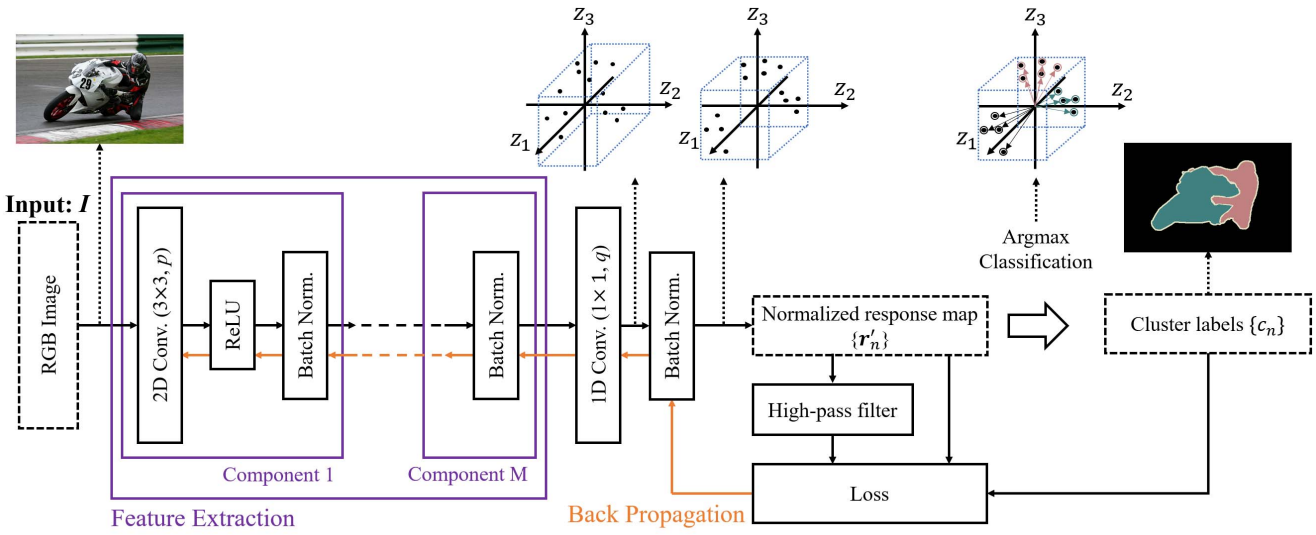


Fig. 1. Illustration of the proposed algorithm for training a CNN. Input image I is fed into the CNN to extract deep features $\{\boldsymbol{x}_n\}$ using a feature extraction module. Subsequently, one-dimensional (1D) convolutional layer calculates the response vectors $\{\boldsymbol{r}_n\}$ of the features in q-dimensional cluster space, where $q = 3$ in this illustration. Here, z_1, z_2, and z_3 represent the three axes of the cluster space. Subsequently, the response vectors are normalized across the axes of the cluster space using a batch normalization function. Further, cluster labels $\{c_n\}$ are determined by assigning the cluster IDs to the response vectors using an argmax function. The cluster labels are then used as pseudo targets to compute the feature similarity loss. Finally, the spatial continuity loss as well as the feature similarity loss are computed and backpropagated.

The concept of loss function for solving criteria (a) and (b) is presented in Section III-B. The details of training a CNN using backpropagation are described in Sec. III-C.

A. Network Architecture

1) Constraint on Feature Similarity: We consider the first criterion of assigning the same label to pixels with similar characteristics. The proposed solution is to apply a linear classifier that classifies the features of each pixel into q classes. In this study, we assume the input to be an RGB image $\mathcal{I} = \{\boldsymbol{v}_n \in \mathbb{R}^3\}$, where each pixel value is normalized to [0, 1]. A p-dimensional feature map $\{\boldsymbol{x}_n\}$ is computed from $\{\boldsymbol{v}_n\}$ through M convolutional components, each of which consists of a two-dimensional (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. Notably, these components for feature extraction can be replaced by alternatives such as fully convolutional networks (FCN) [20]. Subsequently, a response map $\{\boldsymbol{r}_n = W_c \boldsymbol{x}_n\}$ is obtained by applying a linear classifier, where $W_c \in \mathbb{R}^{q \times p}$. The response map is then normalized to $\{\boldsymbol{r}'_n\}$ such that $\{\boldsymbol{r}'_n\}$ has zero mean and unit variance. The motivation behind the normalization process is described in Sec. III-A2. Finally, the cluster label c_n for each pixel is obtained by selecting the dimension that has the maximum value in $\boldsymbol{r}'_n$. This classification rule is referred to as the argmax classification. This processing corresponds intuitively to the clustering of feature vectors into q clusters. The ith cluster of the final responses $\{\boldsymbol{r}'_n\}$ can be written as:

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

where $r'_{n,i}$ denotes the ith element of $\boldsymbol{r}'_n$. 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. Notably, C_i can be $\emptyset$, and therefore the number of unique cluster labels can arbitrarily range from 1 to q.

2) Constraint on the Number of Unique Cluster Labels: In 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. III-A1, the proposed strategy is to classify pixels into an arbitrary number q' ($1 \leq q' \leq q$) of clusters, where q is the maximum possible value of q'. A large q' indicates oversegmentation, whereas a small q' indicates undersegmentation. To train a neural network, we set a large number to the initial (maximum) number of cluster labels q. Then, in the iterative update process, similar or spatially close pixels are integrated by considering feature similarity and spatial continuity constraints. This phenomenon leads to reduce the number of unique cluster labels q', even though there is no explicit constraint on q'.

As shown in Fig. 1, the proposed clustering function based on argmax classification corresponds to q'-class clustering, where q' anchors correspond to a subset of q points at infinity on the q axes. The aforementioned criteria (a) and (b) only facilitate the grouping of pixels, which could lead to a simple solution that $q' = 1$. To prevent this kind of undersegmentation failure, a third criterion (c) is introduced, which is the preference for a large q'. The proposed solution is to insert the *intra-axis* normalization process for the response map $\{\boldsymbol{r}_n\}$ before assigning cluster labels using the argmax classification. Here, batch normalization [46] is used. This operation, also known as whitening, converts the original responses $\{\boldsymbol{r}_n\}$ to $\{\boldsymbol{r}'_n\}$, where each axis has zero mean and unit variance. This gives each $r'_{n,i}$ $(i = 1, \ldots, q)$ an even chance to be the maximum value of $\boldsymbol{r}'_n$ across the axes. Although this operation does not

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

B. Loss Function

The proposed loss function L consists of a constraint on feature similarity and a constraint on spatial continuity, denoted as follows:

$$L = \underbrace{L_{\text{sim}}(\{\boldsymbol{r}'_n, c_n\})}_{\text{feature similarity}} + \mu \underbrace{L_{\text{con}}(\{\boldsymbol{r}'_n\})}_{\text{spatial continuity}}, \tag{1}$$

where μ represents the weight for balancing the two constraints. Although the proposed method is a completely unsupervised learning method, the use of the proposed method with scribbles as user input was also investigated. In the case with segmentation using scribble information, the loss function (1) is simply modified using another weight ν as follows:

$$L = \underbrace{L_{\text{sim}}(\{\boldsymbol{r}'_n, c_n\})}_{\text{feature similarity}} + \mu \underbrace{L_{\text{con}}(\{\boldsymbol{r}'_n\})}_{\text{spatial continuity}} + \nu \underbrace{L_{\text{scr}}(\{\boldsymbol{r}'_n, s_n, u_n\})}_{\text{scribble information}}. \tag{2}$$

Each component of the abovementioned function is described in their respective sections below.

1) Constraint on Feature Similarity: As described in Sec. III-A1, the cluster labels $\{c_n\}$ are obtained by applying the argmax function to the normalized response map $\{\boldsymbol{r}'_n\}$. The cluster labels are further utilized as the pseudo targets. In the proposed approach, the following cross entropy loss between $\{\boldsymbol{r}'_n\}$ and $\{c_n\}$ is calculated as the constraint on feature similarity:

$$L_{\text{sim}}(\{\boldsymbol{r}'_n, c_n\}) = \sum_{n=1}^{N} \sum_{i=1}^{q} -\delta(i - c_n) \ln r'_{n,i},$$

where

$$\delta(t) = \begin{cases} 1 & \text{if } t = 0 \\ 0 & \text{otherwise.} \end{cases}$$

The objective behind this loss function is to *enhance* the similarity of the similar features. Once the image pixels are clustered based on their features, the feature vectors within the same cluster should be similar to each other, and the feature vectors from different clusters should be different from each other. Through the minimization of this loss function, the network weights are updated to facilitate extraction of more efficient features for clustering.

2) Constraint on Spatial Continuity: The elementary concept of image pixel clustering is to group similar pixels into clusters, as shown in Sec. III-A1. In image segmentation, however, it is preferable for the clusters of image pixels to be spatially continuous. An additional constraint is introduced that favors the cluster labels to be the same as those of the neighboring pixels.

In a similar manner to [47], we consider the L1-norm of horizontal and vertical differences of the response map $\{\boldsymbol{r}'_n\}$ as a spatial constraint. We can implement the process by a differential operator. More specifically, the spatial continuity loss L_{con} is defined as follows:

$$L_{\text{con}}(\{\boldsymbol{r}'_n\}) = \sum_{\xi=1}^{W-1} \sum_{\eta=1}^{H-1} || \boldsymbol{r}'_{\xi+1,\eta} - \boldsymbol{r}'_{\xi,\eta} ||_1 + || \boldsymbol{r}'_{\xi,\eta+1} - \boldsymbol{r}'_{\xi,\eta} ||_1,$$

where W and H represent the width and height of an input image, whereas $\boldsymbol{r}'_{\xi,\eta}$ represents the pixel value at (ξ, η) in the response map $\{\boldsymbol{r}'_n\}$.

By applying the spatial continuity loss L_{con}, an excessive number of labels due to complicated patterns or textures can be suppressed.

3) Constraint on Scribbles as User Input: Image segmentation technique with scribble information has been researched extensively [15], [31]–[33]. In the proposed approach, scribble loss L_{scr} as partial cross entropy was introduced as follows:

$$L_{\text{scr}}(\{\boldsymbol{r}'_n, s_n, u_n\}) = \sum_{n=1}^{N} \sum_{i=1}^{q} -u_n \delta(i - s_n) \ln r'_{n,i},$$

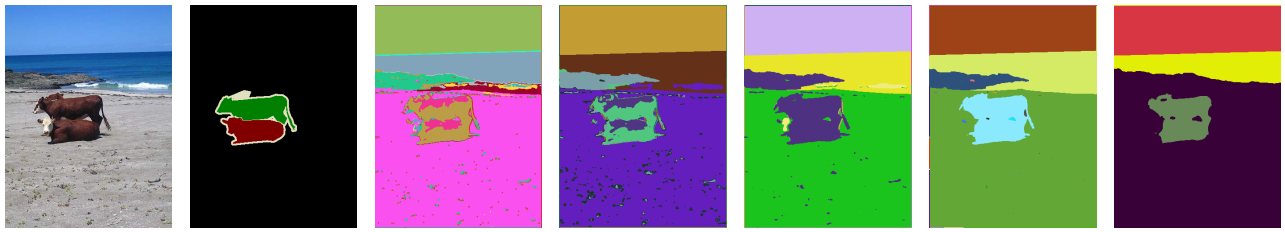
where $u_n = 1$ if the nth pixel is a scribbled pixel, otherwise it is 0, and s_n denotes the scribble label for each pixel.

Algorithm 1 Unsupervised Image Segmentation

Input: $\mathcal{I} = \{\boldsymbol{v}_n \in \mathbb{R}^3\}$ // RGB image
μ // weight for L_{con}
Output: $\mathcal{L} = \{c_n \in \mathbb{Z}\}$ // Label image
$\{W_m, \boldsymbol{b}_m\}_{m=1}^{M} \leftarrow$ `Init()` // Initialize
$W_c \leftarrow$ `Init()` // Initialize
for $t = 1$ **to** T **do**
  $\{\boldsymbol{x}_n\} \leftarrow$ `GetFeats(` $\{\boldsymbol{v}_n\}, \{W_m, \boldsymbol{b}_m\}_{m=1}^{M}$ `)`
  $\{\boldsymbol{r}_n\} \leftarrow \{\ W_c \boldsymbol{x}_n\ \}$
  $\{\boldsymbol{r}'_n\} \leftarrow$ `Norm(` $\{\boldsymbol{r}_n\}$ `)` // Batch norm.
  $\{c_n\} \leftarrow \{\ \arg\max_i r'_{n,i}\ \}$ // Assign labels
  $L \leftarrow L_{\text{sim}}(\{\boldsymbol{r}'_n, c_n\}) + \mu L_{\text{con}}(\{\boldsymbol{r}'_n\})$
  $\{W_m, \boldsymbol{b}_m\}_{m=1}^{M}, W_c \leftarrow$ `Update(` L `)`

C. Learning Network by Backpropagation

In this section, the method of training the network for unsupervised image segmentation is described. Once a target image is input, the following two sub-problems are solved: the prediction of cluster labels with fixed network parameters and the training of network parameters with the (fixed) predicted cluster labels. The former corresponds to the forward process of a network followed by the proposed architecture described in Sec. III-A. The latter corresponds to the backward process of a network based on the gradient descent. Subsequently, we calculate and backpropagate the loss L described in Sec. III-B to update the parameters of the convolutional filters $\{W_m\}_{m=1}^{M}$ as well as the parameters of the classifier W_c. In this study, a stochastic gradient descent with momentum is used for updating the parameters. The parameters are initialized with the Xavier initialization [48], which samples values from the uniform distribution normalized according to the input and output layer size. This forward-backward process is iterated T times to obtain the final prediction of the cluster



(a) Input image (b) Ground truth (c) $\mu = 1$ (d) $\mu = 5$ (e) $\mu = 10$ (f) $\mu = 50$ (g) $\mu = 100$

Fig. 2. Effect of continuity loss with different μ values. Different segments are shown in different colors.

labels $\{c_n\}$. Algorithm 1 shows the pseudocode for the proposed unsupervised image segmentation algorithm. Since this iterative process requires a little computation time, we further introduce a use of the proposed method with one or several reference images. Provided that a target image is somewhat similar to the reference images, the fixed network weights trained with those images as pre-processing can be reused. The effectiveness of the use of reference images is investigated in Sec. IV-C.

As shown in Fig. 1, the proposed CNN network is composed of basic functions. The most distinctive 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 to obtain reasonable labels $\{c_n\}$ (see Sec. III-A2). Furthermore, 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 the parameter updates and clustering, which leads to different solutions of $\{c_n\}$. We set the learning rate to 0.1 with a momentum of 0.9.

IV. Experimental Results

As mentioned in Sec. I, a spatial continuity loss is proposed as described in Sec. III-B2 as an alternative to superpixel extraction used in our previous study [3]. The effectiveness of the continuity loss was evaluated by comparing it with [3] as well as other classical methods discussed in Sec. IV-A. Additionally, the use of the proposed method with scribble input in Sec. IV-B and with reference images in Sec. IV-C was demonstrated. The number of convolutional layers M was set to 3 and $p = q = 100$ for all of the experiments. For the loss function, different μ were set for each experiment: $\mu = 5$ for PASCAL VOC 2012 and BSD500 in Section IV-A and Section IV-C, $\mu = 50$ for iCoseg and BBC Earth in Section IV-C, $\mu = 100$ for pixabay in Section IV-C, and $\mu = 1$ for Section IV-B. The results of all the experiments were evaluated by the mean intersection over union (mIOU). Here, mIOU was calculated as the mean IOU of each segment in the ground truth (GT) and the estimated segment that had the largest IOU with the GT segment. Notably, the object category labels in PASCAL VOC 2012 dataset [49] were ignored and each segment along with the background region was treated as an individual segment.

A. Effect of Continuity Loss

The effect of continuity loss on the validation dataset of PASCAL VOC 2012 segmentation benchmark [49] and Berkeley Segmentation Dataset and Benchmark (BSD500) [51] were evaluated. Figure 2 shows examples of the segmentation results when μ was changed. In case of Fig. 2f, the image was successfully segmented into sky, sea, rock, cattle, and beach regions. However, the image was segmented in more detail with $\mu = 1$; for example, the beach was further segmented into sand and grass regions. It is inferred that the optimal μ changes depending on the degree of detailing in the desired segmentation results. Table II shows the change in mIOU scores with respect to μ and ν variations on PASCAL VOC 2012 dataset [49]. The results show that $\mu = 5$ is the best when applying to unsupervised segmentation and $\nu = 0.5$ is the best for segmentation with user input. It is also shown that the proposed method is more sensitive to ν than μ.

Table I shows comparative results of the unsupervised image segmentation on two benchmark datasets. The k-means clustering and the graph-based segmentation method (GS) [6] were chosen as the comparative methods. In case of GS, a gaussian filter with $\sigma = 1$ was applied to smooth an input image slightly before computing the edge weights, in order to compensate for digitization artifacts. GS needs a threshold parameter to determine the granularity of segments. The threshold parameter effectively sets a scale of observation, in that a larger value causes a preference for larger components. For the k-means clustering, the concatenation of RGB values in a 5×5 window were used for each pixel representation. The connected components were extracted as segments from each cluster generated by k-means clustering and the proposed method. The best k for k-means clustering and threshold parameter τ for GS were experimentally determined from $\{2, 5, 8, 11, 14, 17, 20\}$ and $\{100, 500, 1000, 1500, 2000\}$, respectively. For comparison with a cutting-edge method, we employed Invariant Information Clustering (IIC) [50]. We altered the number of output clusters and iterations as $\{2, 5, 8, 11, 14, 17, 20\}$ and $\{10, 20, 30, 40, 50\}$, respectively, and shows the best result among them. As to other parameters, we used default values used for Microsoft COCO dataset in the official IIC code.[2]

Examples of unsupervised image segmentation results on PASCAL VOC 2012 and BSD500 are shown in Fig. 3 and Fig. 4, respectively. As shown in figure, the boundary lines of segments are smoother and more salient using the proposed

[2]https://github.com/xu-ji/IIC

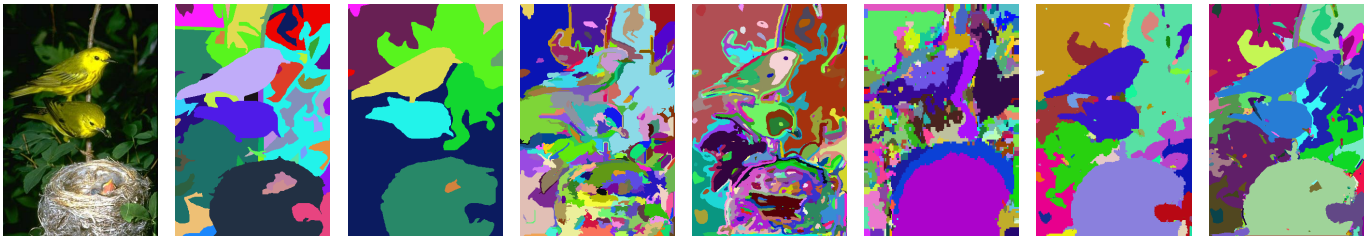
TABLE I

COMPARISON OF MIOU FOR UNSUPERVISED SEGMENTATION ON PASCAL VOC 2012 AND BSD500. THE BEST SCORES ARE SHOWN IN BOLD AND THE SECOND-BEST SCORES ARE UNDERLINED

Method	VOC 2012	BSD500 all	BSD500 fine	BSD500 coarse	mean
k-means clustering, $k = 2$	0.3166	0.1223	0.0865	0.1972	0.1807
k-means clustering, $k = 17$	0.2383	0.2404	0.2208	0.2648	0.2411
Graph-based Segmentation (GS) [6], $\tau = 100$	0.2682	**0.3135**	**0.2951**	0.3255	0.3006
Graph-based Segmentation (GS) [6], $\tau = 500$	**0.3647**	0.2768	0.2238	<u>0.3659</u>	<u>0.3078</u>
IIC [50], $k = 2$	0.2729	0.0896	0.0537	0.1733	0.1474
IIC [50], $k = 20$	0.2005	0.1724	0.1513	0.2071	0.1828
Ours w/ superpixels [3]	0.3082	0.2261	0.1690	0.3239	0.2568
Ours w/ continuity loss, $\mu = 5$	<u>0.3520</u>	<u>0.3050</u>	<u>0.2592</u>	**0.3739**	**0.3225**

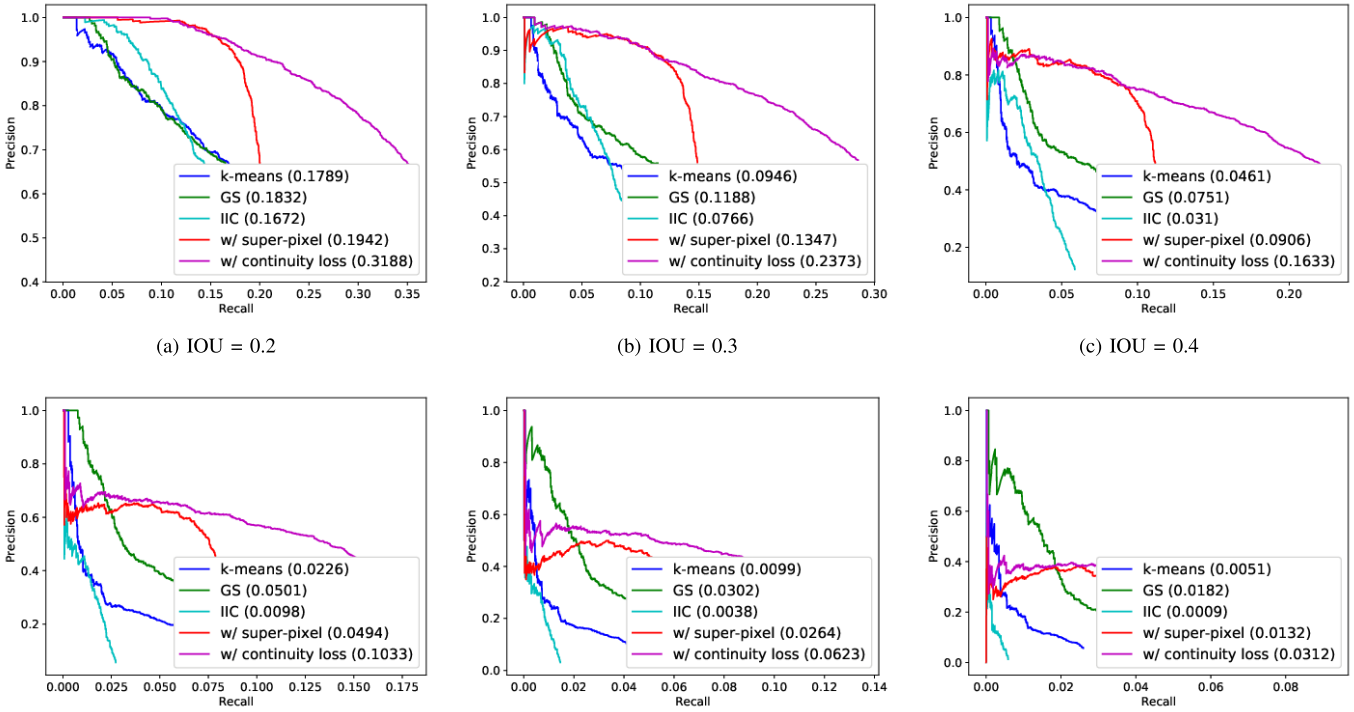
Fig. 3. Comparison of unsupervised segmentation results on PASCAL VOC 2012. The method with superpixels corresponds to the previous method proposed in [3]. Different segments are shown in different colors.

method compared with those in our previous work [3]. This improvement also leads to enhanced performance, which can be confirmed from Table I. There are several sets of ground truth segmentation for each image in the BSD500 superpixel benchmark. Figures 4b and 4c show two different ground truth segments for an exemplar test image. As shown in figure, the ground truth segments are labeled without certain object classes. For evaluation, three groups for mIOU calculation were defined as: "all" using all ground truth files, "fine" using a single ground truth file per image that contains the largest number of segments, and "coarse" using the ground truth file that contains the smallest number of segments. In this case, "fine" used Fig. 4b, "coarse" used Fig. 4c, and "all" used all the ground truth files including both of those for mIOU calculation. According to Table I, the proposed method achieved the best or the second best scores on PASCAL



(a) Input image (b) Ground truth #1 (c) Ground truth #2 (d) GS [6] (e) k-means (f) IIC [50] (g) w/ superpixels [3] (h) w/ continuity loss

Fig. 4. Comparison of unsupervised segmentation results on BSD500. Different segments are shown in different colors. In (b) and (c), two different ground truth segments of image (a) in BSD500 superpixel benchmark are shown.



(a) IOU = 0.2 (b) IOU = 0.3 (c) IOU = 0.4

(d) IOU = 0.5 (e) IOU = 0.6 (f) IOU = 0.7

Fig. 5. Precision-recall curves with different IOU thresholds for BSD500. The numbers in the legends represent the average precision scores of each method.

TABLE II
PARAMETER SEARCH ON PASCAL VOC 2012

unsupervised segmentation						
μ	0.1	0.5	1	5	10	20
mIOU	0.3340	0.3433	0.3449	**0.3520**	0.3483	0.3438
segmentation with user input						
ν	0.1	0.5	1	5	10	20
mIOU	0.4774	**0.6174**	0.5994	0.5298	0.4982	0.4650

TABLE III
ABLATION STUDIES ON L_{con} AND BATCH NORMALIZATION

L_{sim}	L_{con}	BN	VOC2012	BSD500 all	BSD500 fine	BSD500 coarse
✓			0.3312	0.2279	0.1928	0.2932
✓	✓		0.3340	0.2199	0.1832	0.2931
✓		✓	0.3358	0.3007	**0.2619**	0.3506
✓	✓	✓	**0.3520**	**0.3050**	0.2592	**0.3739**

VOC 2012 and BSD500 datasets. The proposed method was outperformed by GS on "BSD500 all" and "BSD500 fine" because the IOU values for small segments are dominant owing to the several small segments in the ground truth sets. This in effect does not convey that the proposed method produced fewer accurate segments than GS. To confirm this fact, the precision-recall curves in "BSD500 all" with an IOU threshold 0.2, 0.3, 0.4, 0.5, 0.6, and 0.7 in Fig. 5 were also presented. For this evaluation, we first sort all the estimated segments according to maximum IOU values between respective estimated segments and the ground truth segments. The precision-recall curves in Fig. 5 are drawn by counting an estimated segment as a true positive when

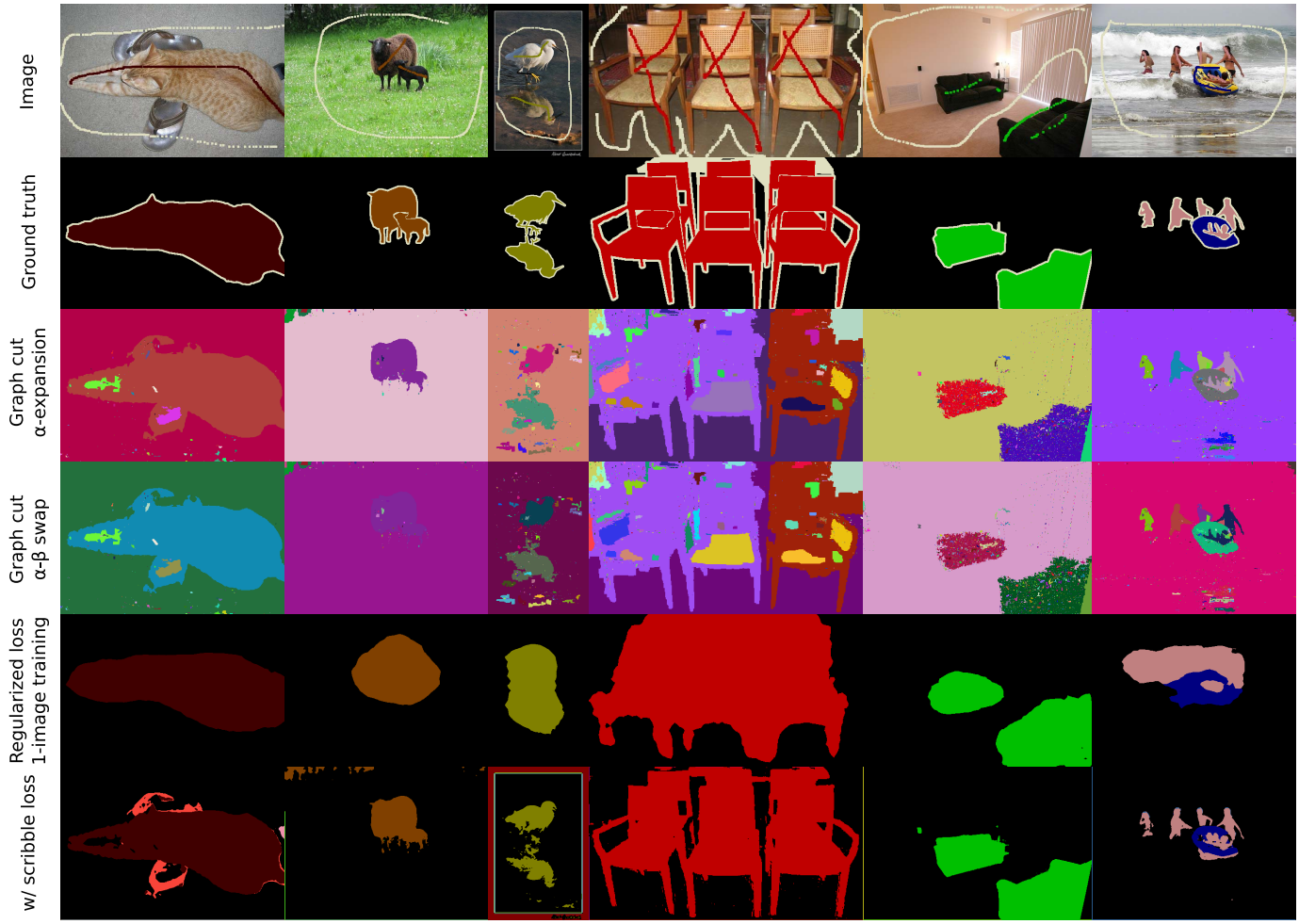


Fig. 6. Comparison of segmentation results with user input. We used DeepLab-ResNet-101 for the base architecture of "Regularized loss 1-image training". "Image" row shows input images including scribbles (user input), which are made bold for the purpose of visualization. Different segments are shown in different colors.

the maximum IOU between the estimated segment and the ground truth segments exceeds a threshold. The number of true positive segments reduces when the threshold increases, which causes different average precision scores in the plots in Fig. 5.

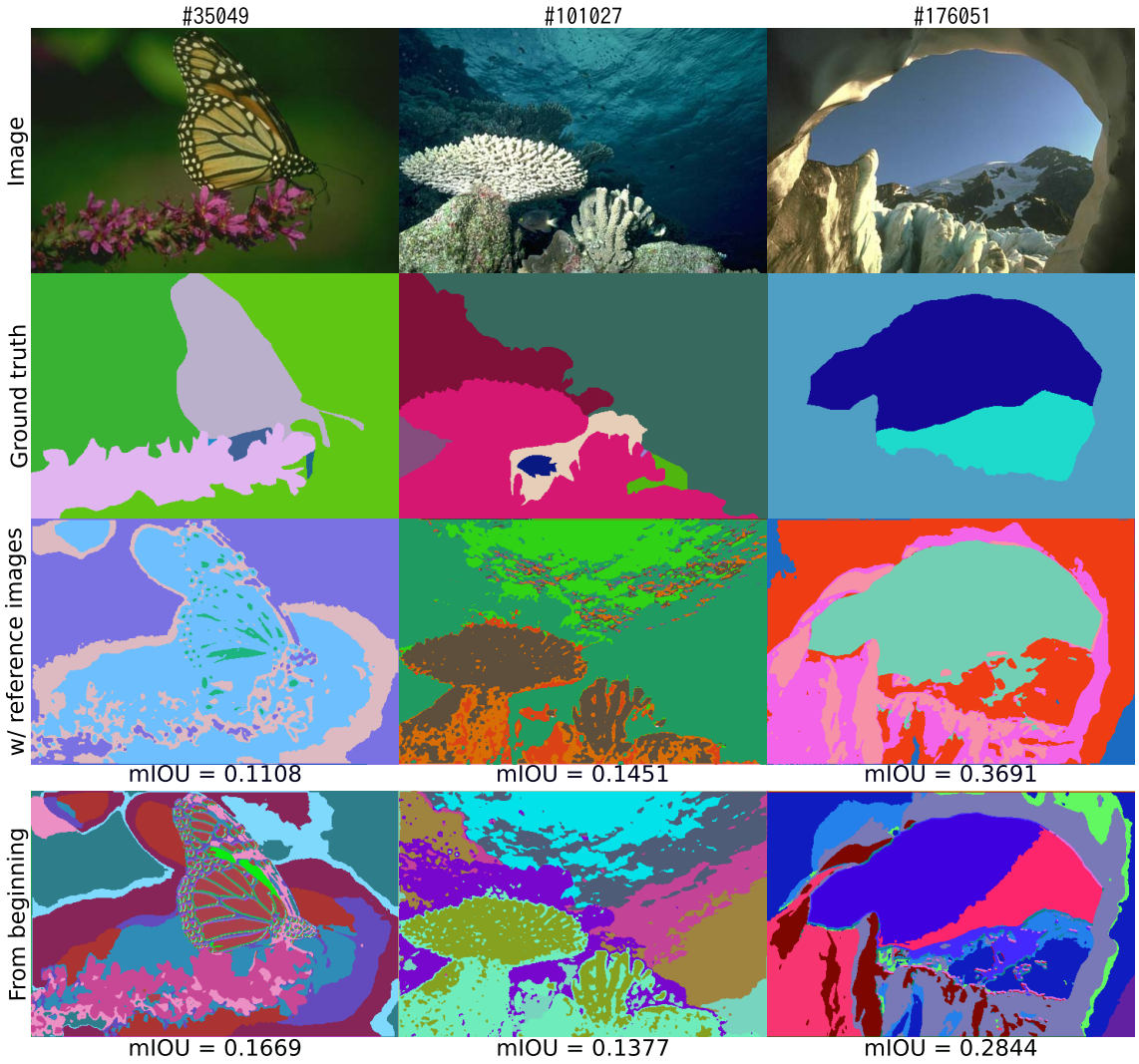
The proposed method w/ continuity loss achieved the best average precision scores and our previous method w/ super-pixels [3] achieved the second best average precision scores for all the cases in Fig. 5.

To confirm the effectiveness of each element of the proposed method, the ablation study was performed with PASCAL VOC 2012 and BSD500. Table III shows the results regarding the presence and absence of L_{con} and the batch normalization of the response map. The experimental results show that the batch normalization process consistently and considerably improves the performance on all the datasets. Even though the effect of L_{con} alone is marginal, it gives a solid improvement when used together with the batch normalization. This indicates the importance of the three criteria introduced in Sec. I.

B. Segmentation With Scribbles as User Input

The effect of the proposed method was tested for image segmentation with the user input on the validation dataset of PASCAL VOC 2012 segmentation benchmark [49]. We let $\nu = 0.5$ in (2) in this experiment. The scribble information was used for the test images given in [31] as the user input. For comparison, graph cut [17], graph cut α-expansion [17], graph cut α-β swap [17], and regularized loss [33] were employed. In graph cut, Gaussian Mixture Model (GMM) was used for modeling foreground and background of an image. A graph is constructed from pixel distributions modeled by GMM. At this time, scribbled pixels are fixed to their scribbled labels which are foreground or background. In the generated graph, a node is defined as a pixel, whereas the weight of an edge connecting nodes is defined by a probability to be foreground or background. Thereafter, the graph is divided by energy minimization into the two groups: foreground and background. The vanilla graph cut is an algorithm for segmenting the foreground and the background and it does not support the multi-labels case. Therefore, in this study, a graph cut was performed multiple times where each scribble is regarded as the foreground each time, and subsequently all the extracted segments were used for calculating the mIOU. To compare the performance, α-expansion and α-β swap (introduced in Sec. II-A), as well as regularized loss [33] were tested. Regularized loss [33] is a weakly-supervised method of segmentation using a training data set and additional scribble

(a) Reference images



(b) Test results

Fig. 7. Results of segmentation with reference images on BSD500. Different segments are shown in different colors.

information. In order to unify the experiment conditions, one image from the validation dataset of PASCAL VOC 2012 is used for the network training with the scribble information. The output in the final iteration for the image after completion of training was regarded as the segmentation result of the image. After that, the network weight is initialized, and then the process is repeated for the next image. This process was repeated individually for all the test images in the validation dataset of PASCAL VOC 2012. This process was defined as "Regularized loss 1-image training". We tested two base architectures for "Regularized loss 1-image training": DeepLab-largeFOV and DeepLab-ResNet-101.

Exemplar segmentation results are shown in Fig. 6. It was observed that the proposed method is more stable than the graph-based methods. Relatively rougher segments of objects are detected by "Regularized loss 1-image training", whereas the boundaries of segmented areas of the proposed method are more accurate. The quantitative evaluation in Table IV shows that the proposed method achieved the best mIOU score. In addition to outperforming than "Regularized loss 1-image

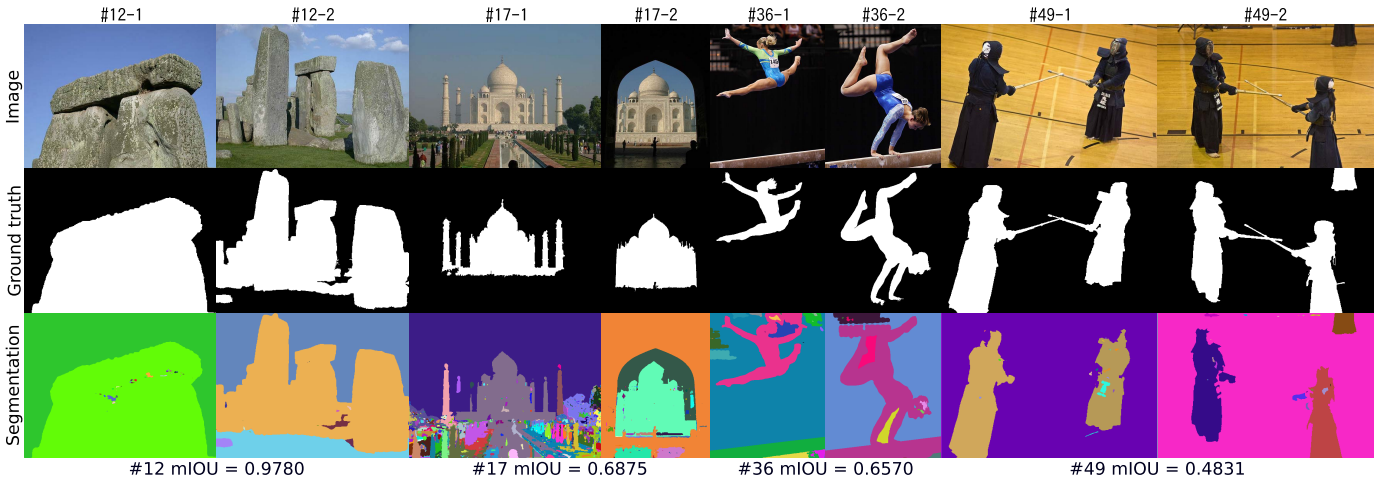


Fig. 8. Results of segmentation with reference images on iCoseg. "#*n*-*m*" denotes the *m*th test image in the group whose ID is *n*. Different segments are shown in different colors.

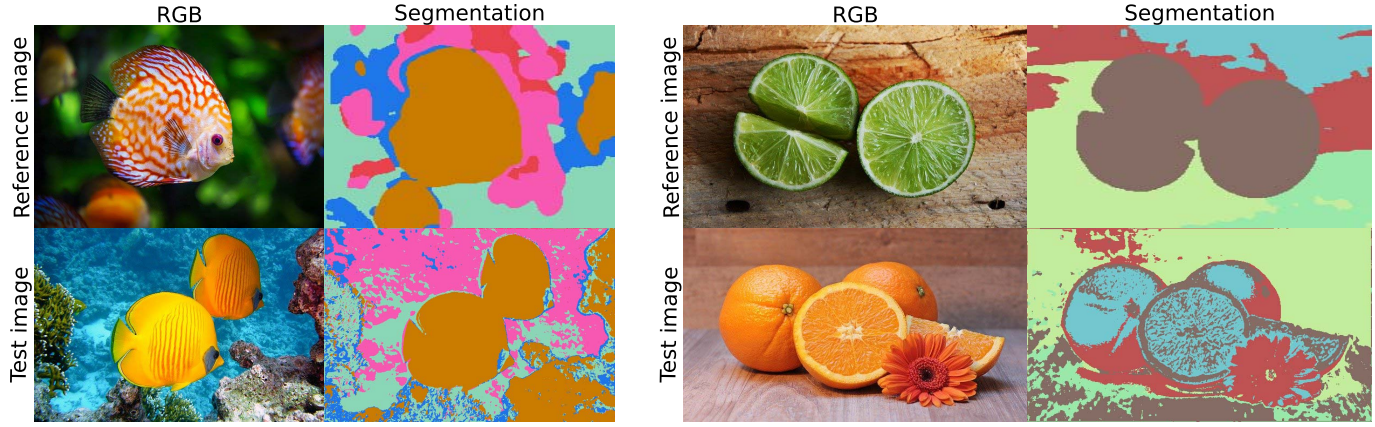


Fig. 9. Results of segmentation with a single reference image. Different segments are shown in different colors. The results imply that similar objects with somewhat similar colors are assigned the same label (see fishes in the left case), although otherwise not (see oranges in the right case). Images are from pixabay [52].

TABLE IV
COMPARISON OF THE NUMBER OF PARAMETERS, COMPUTATION TIME, AND MIOU FOR SEGMENTATION WITH USER INPUT

Method	# parameters	Time (sec.)	mIOU
Graph cut [17]	-	1.47	0.2965
Graph cut α-expansion [17]	-	0.81	0.5509
Graph cut α-β swap [17]	-	0.77	0.5524
Regularized loss [33] 1-image training (DeepLab-largeFOV)	20,499,136	42	0.5790
Regularized loss [33] 1-image training (DeepLab-ResNet-101)	132,145,344	414	0.6064
Proposed method w/ scribble loss	103,600	20	**0.6174**

training" with the DeepLab-ResNet-101 architecture, the proposed method is effective in three folds. First, the proposed method uses a small network where the number of parameters is 1,000 times less than DeepLab-ResNet-101. Owing to the smallness of the architecture, the proposed method converges 20 times faster than "Regularized loss 1-image training" with the DeepLab-ResNet-101 architecture. Finally, the proposed method initializes the network with random weights and thus requires no pre-trained weights. In contrast, "Regularized loss 1-image training" requires the weights pre-trained on *e.g.*, the ImageNet dataset[3] for initialization. Notably, we found that "Regularized loss 1-image training" both with the DeepLab-ResNet-101 and DeepLab-largeFOV architectures failed to train the weights from random states in our experiment.

[3]We used the pre-trained weights downloaded from http://liangchiehchen.com/projects/Init%20Models.html and https://github.com/KaimingHe/deep-residual-networks for DeepLab-largeFOV and DeepLab-ResNet-101, respectively.

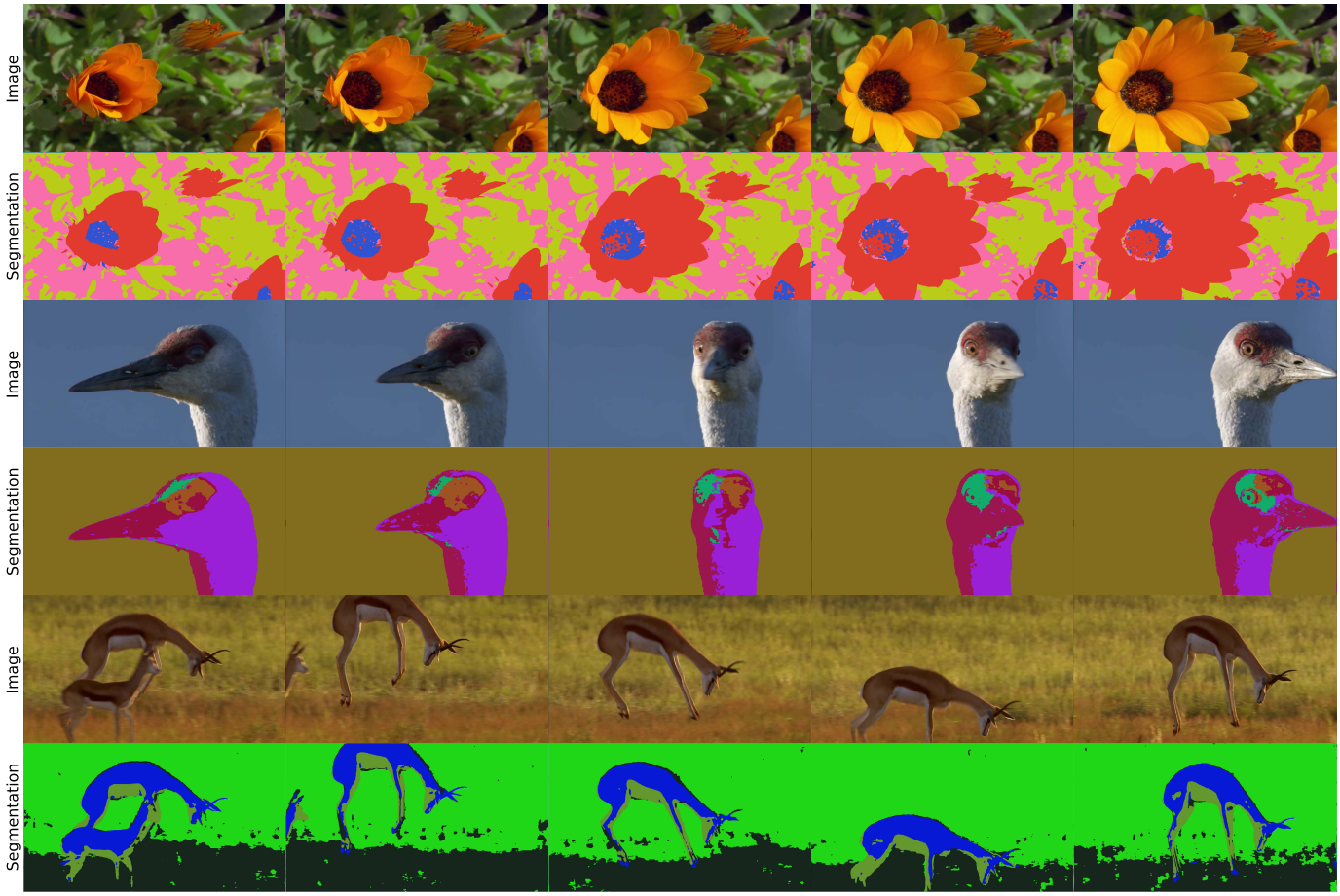


Fig. 10. Results of segmentation for sequential images. For each case, a network is trained with only the leftmost image in the respective row. Different segments are shown in different colors and the time flow is from left to right. The images are from "BBC Earth, Nature Makes You Happy." [53].

C. Unsupervised Segmentation With Reference Images

Supervised learning generally learns from training data and evaluates the performance using test data. Therefore, the network can obtain segmentation results by processing the test images with the (fixed) learned weights. In contrast, as the proposed method is completely unsupervised learning, it is necessary to learn the network weights every time the test image is input in order to obtain the segmentation results. Further, an unsupervised segmentation experiment was conducted with reference images. The effectiveness of the networks of fixed weights trained on several images as reference was evaluated for unseen test images. The BSD500 and the iCoseg [54] datasets were employed for the experiment.

The proposed method was trained with four images in BSD500 shown in Fig. 7a. In the training phase, the network was updated once for each reference image. After training, the network weights were fixed and the other three images shown in the top row in Fig. 7b were segmented. The reference images and the test images were arbitrarily selected from different scenes in the nature category. The segmentation results are shown in the two bottom rows in Fig. 7b. The phrase "from beginning" in Fig. 7b means that an image is segmented with the proposed method where the weights of a network are trained for each test image from scratch. As shown in Fig. 7b, the segmentation results "w/ reference images" were more detailed than "from beginning". This is because "from beginning" integrates clusters under the influence of the continuity loss when training the target image. According to Fig. 7b, "w/ reference images" showed acceptable segmentation performance compared with "from beginning". The method "w/ reference images" only takes under $0.02s$ for the segmentation of each image, whereas the "from beginning" method takes approximately $20s$ under GPU calculation in GeForce GTX TITAN X to get the segmentation results. The proposed method with four groups in iCoseg (ID: 12, 17, 36, 49) were also trained. As iCoseg does not distinguish between the training and test data, two images from the group were randomly selected for testing. Further, the proposed method was trained on the images in the group excluding the sampled test images. The segmentation results are illustrated in Fig. 8. Therefore, it was concluded that it is possible to segment unknown images with unsupervised trained weights on reference images, provided that the images are somewhat similar to the reference images (*e.g.*, when they belong to the same category).

We also conducted an experiment to segment an image using a single reference image. Figure 9 shows the segmentation results of the test and reference images. Even though the

segmentation result for a test image is not as appropriate as that of a reference image, a sufficient segmentation result is obtained. We can see different levels of quality in these two cases: fishes in the left case are successfully assigned the same label, whereas oranges in the right case are differentiated. It implies that similar objects with somewhat similar colors are assigned the same label.

In the experiments thus far, it was found that the proposed method can be trained from several reference images and effective to similar-different images. Therefore, another application for videos was introduced. Video data generally contains information connected in a time series. Hence, video segmentation can be accomplished by training only a part of all the frames using the proposed method. Figure 10 shows examples of segmentation results when video data was input in the proposed method. The proposed method trained a network only with the leftmost image in a respective row in Fig. 10. It was observed that most of the segments obtained from other images were successfully matched to the same segments in the leftmost image. Consequently, it was demonstrated that even the video data without ground truth can be segmented with the proposed method efficiently using only a single frame as a reference. This result indicates that the proposed method, which aims unsupervised learning of image segmentation, can be extended to unsupervised learning of video segmentation. By using the first frame of the video as a reference and segmenting other frames, the segmentation task can be accelerated. In addition, the segmentation of the full target video can also be improved by stacking processed images as additional reference images.

V. Conclusion

A novel CNN architecture was presented in this study, along with its unsupervised process that enables image segmentation in an unsupervised manner. The proposed CNN architecture consists of convolutional filters for feature extraction and differentiable processes for feature clustering, which enables end-to-end network training. The proposed CNN jointly assigned cluster labels to image pixels and updated the convolutional filters to achieve better separation of clusters using the backpropagation of the proposed loss to the normalized responses of convolutional layers. Furthermore, two applications based on the proposed segmentation method were introduced: segmentation with scribbles as user input and utilization of reference images. The experimental results on the PASCAL VOC 2012 segmentation benchmark dataset [49] and BSD500 [51] demonstrated the effectiveness of the proposed method for completely unsupervised segmentation. The proposed method outperformed classical methods for unsupervised image segmentation such as k-means clustering and a graph-based segmentation method, which verfied the importance of feature learning. Furthermore, the effectiveness of the proposed method for image segmentation with user input and utilization of reference images was validated by additional experimental results on the PASCAL VOC 2012, BSD500, and iCoseg [54] datasets. A potential application of the proposed method to an efficient video segmentation system was demonstrated.

References

[1] R. Unnikrishnan, C. Pantofaru, and M. Hebert, "Toward objective evaluation of image segmentation algorithms," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 6, pp. 929–944, Jun. 2007.

[2] A. Y. Yang, J. Wright, Y. Ma, and S. S. Sastry, "Unsupervised segmentation of natural images via lossy data compression," *Comput. Vis. Image Understand.*, vol. 110, no. 2, pp. 212–225, May 2008.

[3] A. Kanezaki, "Unsupervised image segmentation by backpropagation," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 1543–1547.

[4] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC superpixels compared to State-of-the-Art superpixel methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2274–2282, Nov. 2012.

[5] J. MacQueen *et al.*, "Some methods for classification and analysis of multivariate observations," in *Proc. 5th Berkeley Symp. Math. Statist. Probab.*, vol. 1, Oakland, CA, USA, 1967, pp. 105–117.

[6] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *Int. J. Comput. Vis.*, vol. 59, no. 2, pp. 167–181, Sep. 2004.

[7] X. Liu, Q. Xu, J. Ma, H. Jin, and Y. Zhang, "MsLRR: A unified multiscale low-rank representation for image segmentation," *IEEE Trans. Image Process.*, vol. 23, no. 5, pp. 2159–2167, May 2014.

[8] X. Xia and B. Kulis, "W-net: A deep model for fully unsupervised image segmentation," 2017, *arXiv:1711.08506*. [Online]. Available: http://arxiv.org/abs/1711.08506

[9] I. Croitoru, S.-V. Bogolin, and M. Leordeanu, "Unsupervised learning of foreground object segmentation," *Int. J. Comput. Vis.*, vol. 127, no. 9, pp. 1279–1302, Sep. 2019.

[10] S. Ghosh, N. Das, I. Das, and U. Maulik, "Understanding deep learning techniques for image segmentation," *ACM Comput. Surveys*, vol. 52, no. 4, pp. 1–35, Sep. 2019.

[11] Y. Y. Boykov and M.-P. Jolly, "Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images," in *Proc. 8th IEEE Int. Conf. Comput. Vis.*, Jul. 2001, pp. 105–112.

[12] V. Lempitsky, P. Kohli, C. Rother, and T. Sharp, "Image segmentation with a bounding box prior," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep. 2009, pp. 277–284.

[13] A. Levin, D. Lischinski, and Y. Weiss, "A closed-form solution to natural image matting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 2, pp. 228–242, Feb. 2008.

[14] A. Levin, A. Rav-Acha, and D. Lischinski, "Spectral matting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 10, pp. 1699–1712, Oct. 2008.

[15] W. Yang, J. Cai, J. Zheng, and J. Luo, "User-friendly interactive image segmentation through unified combinatorial user inputs," *IEEE Trans. Image Process.*, vol. 19, no. 9, pp. 2470–2479, Sep. 2010.

[16] E. Zemene and M. Pelillo, "Interactive image segmentation using constrained dominant sets," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2016, pp. 278–294.

[17] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," in *Proc. 7th IEEE Int. Conf. Comput. Vis.*, 1999, pp. 1–7.

[18] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, Dec. 2017.

[19] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected CRFs," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2015, pp. 1–4.

[20] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.

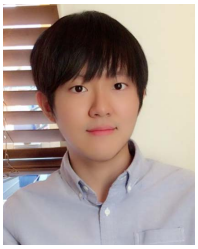
[21] S. Zheng *et al.*, "Conditional random fields as recurrent neural networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1527–1537.

[22] J. Tighe and S. Lazebnik, "Finding things: Image parsing with regions and per-exemplar detectors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 3001–3008.

[23] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik, "Simultaneous detection and segmentation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2014, pp. 293–317.

[24] J. Dai, K. He, and J. Sun, "Instance-aware semantic segmentation via multi-task network cascades," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3150–3158.

[25] J. Zhu, J. Mao, and A. L. Yuille, "Learning from weakly supervised data by the expectation loss svm (E-SVM) algorithm," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2014, pp. 1125–1133.
[26] F.-J. Chang, Y.-Y. Lin, and K.-J. Hsu, "Multiple structured-instance learning for semantic segmentation with uncertain training data," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 360–367.
[27] D. Pathak, P. Krahenbuhl, and T. Darrell, "Constrained convolutional neural networks for weakly supervised segmentation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1796–1804.
[28] N. Pourian, S. Karthikeyan, and B. S. Manjunath, "Weakly supervised graph based semantic segmentation by learning communities of image-parts," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1359–1367.
[29] Z. Shi, Y. Yang, T. M. Hospedales, and T. Xiang, "Weakly-supervised image annotation and segmentation with objects and attributes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2525–2538, Dec. 2017.
[30] W. Shimoda and K. Yanai, "Distinct class-specific saliency maps for weakly supervised semantic segmentation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2016, pp. 218–234.
[31] D. Lin, J. Dai, J. Jia, K. He, and J. Sun, "ScribbleSup: Scribble-supervised convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3159–3167.
[32] M. Tang, A. Djelouah, F. Perazzi, Y. Boykov, and C. Schroers, "Normalized cut loss for weakly-supervised CNN segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Dec. 2018, pp. 1818–1827.
[33] M. Tang, F. Perazzi, A. Djelouah, I. Ben Ayed, C. Schroers, and Y. Boykov, "On regularized losses for weakly-supervised CNN segmentation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 507–522.
[34] J. Carreira and C. Sminchisescu, "CPMC: Automatic object segmentation using constrained parametric min-cuts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 7, pp. 1312–1328, Jul. 2012.
[35] Y. Zhang and T. Chen, "Efficient inference for fully-connected CRFs with stationarity," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 582–589
[36] Z. Huang, X. Wang, J. Wang, W. Liu, and J. Wang, "Weakly-supervised semantic segmentation network with deep seeded region growing," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7014–7023.
[37] H. Lee, P. Pham, Y. Largman, and A. Y. Ng, "Unsupervised feature learning for audio classification using convolutional deep belief networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2009, pp. 1047–1096.
[38] Q. V. Le, "Building high-level features using large scale unsupervised learning," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2013, pp. 8595–8598.
[39] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations," in *Proc. 26th Annu. Int. Conf. Mach. Learn.*, 2009, pp. 609–616.
[40] M. Cimpoi, S. Maji, and A. Vedaldi, "Deep filter banks for texture recognition and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3828–3836.
[41] B. Hariharan, P. Arbelaez, R. Girshick, and J. Malik, "Hypercolumns for object segmentation and fine-grained localization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 447–456.
[42] J. Xie, R. Girshick, and A. Farhadi, "Unsupervised deep embedding for clustering analysis," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2016, pp. 478–487.
[43] L. Xu, J. Neufeld, B. Larson, and D. Schuurmans, "Maximum margin clustering," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2005, pp. 1537–1547.
[44] F. R. Bach and Z. Harchaoui, "Diffrac: A discriminative and flexible framework for clustering," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2008, pp. 49–56.
[45] A. Joulin, F. Bach, and J. Ponce, "Discriminative clustering for image co-segmentation," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 1943–1950.
[46] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2015, pp. 1–10.
[47] T. Shibata, M. Tanaka, and M. Okutomi, "Misalignment-robust joint filter for cross-modal image pairs," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 3295–3304.
[48] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. Int. Conf. Artif. Intell. Statist. (AISTATS)*, 2010, pp. 249–256.
[49] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes challenge: A retrospective," *Int. J. Comput. Vis.*, vol. 111, no. 1, pp. 98–136, Jan. 2015.
[50] X. Ji, A. Vedaldi, and J. Henriques, "Invariant information clustering for unsupervised image classification and segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9865–9874.
[51] P. Arbeláez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 898–916, May 2011.
[52] Pixabay. *Internet*. Accessed: May 23, 2020. [Online]. Available: https://pixabay.com/
[53] B. Earth. (2017). *Nature Makes You Happy*. Accessed: 2017. [Online]. Available: https://youtu.be/1wkPMUZ9vX4
[54] D. Batra, A. Kowdle, D. Parikh, J. Luo, and T. Chen, "ICoseg: Interactive co-segmentation with intelligent scribble guidance," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 3169–3176.



Wonjik Kim (Member, IEEE) received the Bachelor of Engineering degree from the Department of Control and Systems Engineering, Tokyo Institute of Technology, in 2018, and the Master of Engineering degree from the Department of Systems and Control Engineering, Tokyo Institute of Technology, in 2020, where he is currently pursuing the Ph.D. degree. He has also been a Research Assistant with Artificial Intelligent Research Center, National Institute of Advanced Industrial Science and Technology in 2018.

Asako Kanezaki (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in information science and technology from The University of Tokyo in 2008, 2010, and 2013, respectively. In 2010, she was a Visiting Researcher with Intelligent Autonomous Systems Group, Technische Universität München. From 2013 to 2016, she was an Assistant Professor with The University of Tokyo. She was a Researcher with Living Intelligence Research Team, National Institute of Advanced Industrial Science and Technology (AIST), from 2016 to 2020, where she has been a Senior Researcher since 2018. Since 2020, she has been an Associate Professor with the School of Computing, Tokyo Institute of Technology.

Masayuki Tanaka (Member, IEEE) received the bachelor's and master's degrees in control engineering and the Ph.D. degree from the Tokyo Institute of Technology in 1998, 2000, and 2003. He was a Software Engineer with Agilent Technology from 2003 to 2004. He was a Research Scientist with the Tokyo Institute of Technology from 2004 to 2008, where he was an Associate Professor with the Graduate School of Science and Engineering from 2008 to 2016. He was a Visiting Scholar with the Department of Psychology, Stanford University, from 2013 to 2014. He was an Associate Professor with the School of Engineering, Tokyo Institute of Technology, from 2016 to 2017. He was a Senior Researcher with the National Institute of Advanced Industrial Science and Technology from 2017 to 2020. Since 2020, he has been an Associate Professor with the School of Engineering, Tokyo Institute of Technology.