Weakly supervised graph based semantic segmentation by learning communities of image-parts

Niloufar Pourian, S. Karthikeyan, and B.S. Manjunath
Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA
npourian@ece.ucsb.edu, karthikeyan@ece.ucsb.edu, manj@ece.ucsb.edu

Abstract

We present a weakly-supervised approach to semantic segmentation. The goal is to assign pixel-level labels given only partial information, for example, image-level labels. This is an important problem in many application scenarios where it is difficult to get accurate segmentation or not feasible to obtain detailed annotations. The proposed approach starts with an initial coarse segmentation, followed by a spectral clustering approach that groups related image parts into communities. A community-driven graph is then constructed that captures spatial and feature relationships between communities while a label graph captures correlations between image labels. Finally, mapping the image level labels to appropriate communities is formulated as a convex optimization problem. The proposed approach does not require location information for image level labels and can be trained using partially labeled datasets. Compared to the state-of-the-art weakly supervised approaches, we achieve a significant performance improvement of 9% *on MSRC-21 dataset and* 11% *on LabelMe dataset, while being more than* 300 *times faster.*

1. Introduction

We consider the problem of semantic segmentation to predict a label for every image pixel. Semantic segmentation benefits a variety of vision applications, such as object recognition, automatic driver assistance and 3D urban modeling. However in practice, two key factors limit the applicability of semantic segmentation: speed and scalability. In this work we develop an effective semantic segmentation method which addresses these issues.

Semantic segmentation approaches can be broadly categorized into fully supervised, weakly supervised and unsupervised methods. There is a wealth of published literature on fully supervised semantic segmentation that rely on location information associated with image labels (fully segmented training data) [3, 5, 7, 8, 9, 17, 18, 20, 23, 25, 26, 29, 31, 36, 40, 41, 42, 45, 49, 50, 55]. The aforementioned approaches require that every pixel in the training set be manually labeled. The work in [8] considers a setting in which the training dataset is a mixture of object segments and a set of objects' bounding boxes. Despite adding more flexibility, [8] still depends on the manual localization of objects by a combination of regional segments and bounding boxes. The approaches of [9, 55] focus on segmentation templates to address the problem of image labeling. Such approaches have computational advantages over pixel-based approaches by constraining the search process, however, they still depend on providing the ground truth of object boundaries in training. Creating annotations is costly and time-consuming, thus limiting the applicability of the aforementioned approaches to small sized datasets.

In order to scale semantic segmentation to larger datasets, weakly supervised semantic segmentation approaches have been proposed [1, 30, 44, 46, 47, 48, 52, 53]. These methods typically require fully annotated image-level labels without relying on location information associated to image labels. These approaches use image-level object annotations to learn priors. Although more flexible than fully supervised methods, these techniques typically assume that one has access to all annotations associated with the image which still limits their practical applicability. Our work is closely related to the weakly supervised approaches while being applicable to both fully/partially labeled datasets.

We note that there are segmentation methods that work with unlabeled data, for example see [51], but they tend to be not very robust given the under-constrained nature of the problem. Label correlation enhance the performance by providing stronger priors and unsupervised approaches are inherently unable to do that.

Several works have utilized multiple segmentations to address the problem of semantic segmentation by either changing the parameters of a bottom-up approach [19, 24], or by using different segmentation methods [10, 21, 34]. To avoid the increased complexity associated with multiple segmentations, our model learns image parts using a single segmentation by jointly considering the visual and spatial

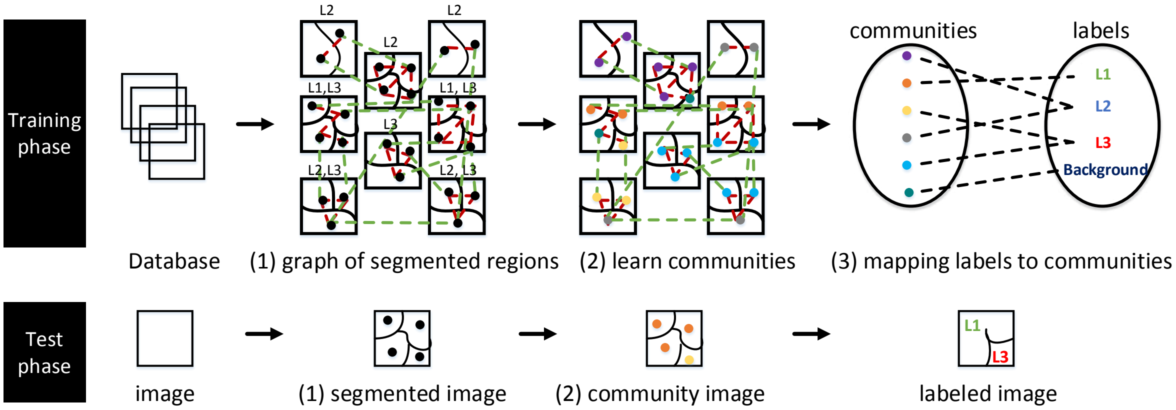

Figure 1: The overall framework of the proposed approach discussed in section 2. This includes the following steps in the reaining phase: (1) graph of segmented regions, (2) community detection, and (3) mapping labels to the learned communities. The test phase consists of (1) segmentation, and (2) mapping regions to the semantically learned communities. L_i with $i = \{1, \dots, 3\}$ indicates the image level labels associated with training images. At test time, mapping regions of a given image to the learned commnunities results in a semantically labeled image.

characteristics of all the images together. This ensures robustness to segmentation variations. In addition, many researchers address the problem of semantic segmentation by finding the labels associated with each pixel or superpixel in the image [40, 48, 52].

In addition, there has been considerable interest in object-based methods for semantic segmentation [2, 14, 27, 33, 43]. In particular, the work in [2, 14] focus on representation and classification of individual regions by designing region-based object detectors and subsequently combining their outputs. This requires generating hierarchical tree of regions at different levels of contrast, and dealing with a large number (i.e. 1000) of generic regions per image which is computationally expensive. Another research area relevant to semantic segmentation is co-segmentation [11, 38, 54]. Here, images which share semantic content are segmented together to extract meaningful common information.

Additionally, several semantic segmentation approaches involve solving a non-convex optimization problem [10, 24, 30, 48]. A solution to a non-convex formulation might only be a local optima and there is no guarantee that the solution is a global one. In particular, [30] involves solving an iterative optimization which is computationally costly and does not necessarily converge to an optimal solution. In contrast, our formulation deals with a convex optimization problem that can be solved efficiently [4].

Main Contributions: Based on image segmentation, we use an attributed graph to capture the visual and spatial characteristics of different image-parts. The communities of related segmented regions among all images are discovered by applying a spectral graph partitioning algorithm to segregate highly related segments. This results in finding meaningful image-part groupings. The relationships among communities and labels are separately modeled. An optimization framework is introduced as a graph learning approach to derive an appropriate semantic for each detected object/community within an image. The overall framework of the proposed approach is depicted in Figure 1. Summarizing,

- Our formulation encodes label co-occurrences and introduces a convex minimization problem for mapping labels to communities. This optimization problem has a closed form solution and can be efficiently solved.
- We learn groups of related image regions and then map higher level semantics to small number of detected communities, eliminating the need for pixel-wise semantic mappings. This makes our method computationally efficient.
- The proposed work is robust to segmentation variations and does not require manual segmentation in training or testing phases. In addition, this method can be applied to databases where image-level labels are only partially provided.

2. Approach

The proposed system learns different parts of an image by grouping related image-parts and labeling each object present in the image with its high level semantics. The details of our approach is as follows.

2.1. Image-Driven Graph

Inspired by [37, 48], we introduce an image-driven graph to capture both visual and spatial characteristics of image regions. This is followed by a graph partitioning algorithm

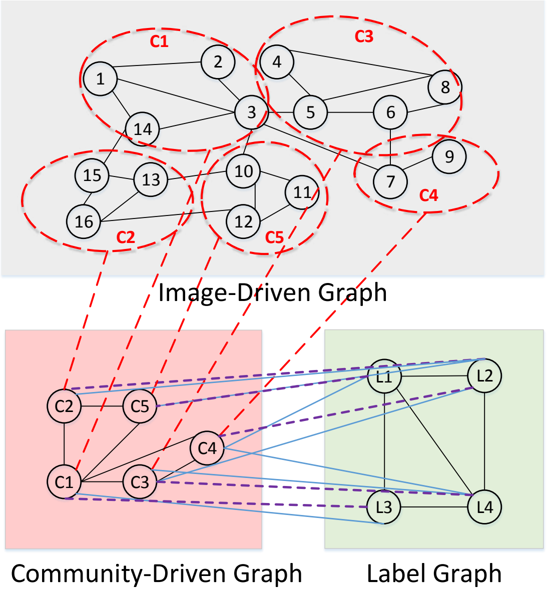

Figure 2: Image-driven graph (top), community-driven graph (bottom left), label graph (bottom right), along with the initial and final label estimations for each of the detected communities. Each node I_i in the fused data graph denotes a segmented region belonging to an image i from the database. Each node c_j in the community-driven graph denotes a detected community in the fused data graph. Each node L_k in the label graph represents a unique label. Each dotted ellipse represents a detected community clustering related nodes (segmented regions) together. The solid and dotted lines between the label graph and community-driven graph represent the initial and final label assignemts to communities, respectively.

to segregate highly related regions across all images in the database.

To provide spatial information, we utilize a segmentation algorithm based on color and texture [13]. We define $G^{(I)} = (V^{(I)}, A^{(I)})$ to be the image-driven graph with $V^{(I)}$ and $A^{(I)}$ representing the nodes and edges, respectively. The image-driven graph $G^{(I)}$ contains $\sum_{d=1}^{\mathcal{D}} |v_d|$ number of nodes, where $|v_d|$ denotes the number of segmented regions of image d, and $\mathcal{D}$ is the total number of images in the database. Two nodes i and j are connected if they are spatially adjacent or if they are visually similar. This is summarized as the following:

$$A_{ij}^{(I)} = \mathcal{I}(i \in \mathcal{F}_j \quad or \quad i \in \mathcal{H}_j), \quad \forall i, j \in V^{(I)} \qquad (1)$$

where $\mathcal{I}(x)$ is an indicator function and is equal to 1 if x holds true and zero otherwise. In addition, $\mathcal{F}_j$ indicates the set of all nodes (segmented regions) that are visually similar to node j and $\mathcal{H}_j$ is the set of all nodes in the spatial neighborhood of node j.

To represent each segmented region, DenseSift features [32] are extracted from each image and then quantized into a visual codebook. To form a regional signature h_i for a node i, features are mapped to the segmented regions that they belong to and a histogram representation of the codebook is constructed.

Two nodes i and j are considered visually similar if their visual similarity score is higher than a threshold $\alpha > 0$. The visual similarity score is defined by the following:

$$\Lambda(h_i, h_j) = \underbrace{e^{-\Delta(h_i,h_j)}}_{regional\ similarity} \underbrace{\mathcal{I}(u_i^T u_j > 0)}_{label\ similarity} \qquad (2)$$

where $\Delta(h_i, h_j)$ represents the distance between the regional representations of node i and j, and u_i denotes a binary label vector of node i with length equal to the total number of labels in the dataset. u_{ij} is equal to one iff node i is associated with label j. Each node is associated with the labels of the image that they belong to. Label similarity limits us to consider visual similarity between nodes that share label(s). The distance between two regional histograms is measured by the Hellinger distance. This is a suitable for computing the distance between histograms in classification and retrieval problems [35].

2.2. Community-Driven Graph

Our goal is to find similar/related image-parts in the graphical image representation. This is done by applying a graph partitioning algorithm to the image-driven graph. We refer to each of these groups as a community. Each community resembles a bag of related image-parts.

For graph partitioning, we use the normalized cut method as described in [39]. In this algorithm, the quality of the partition (cut) is measured by the density of the weighted links inside communities as compared to the weighted links between communities. The objective is to maximize sum of the weighted links within a community while minimizing sum of the weighted links across community pairs.

Suppose graph $G^{(I)}$ consists of $\mathcal{C}$ detected communities. Each community c_i with $i = \{1, \ldots, \mathcal{C}\}$ contains all the pieces/image-parts of an object and by mapping these communities to each segmented image, one can detect any specific object. Figure 3 illustrates some sample images highlighted by a color representing the community that each segmented region belongs to. It is worth noting that if we choose C to be small (fewer number of communities), the detected communities may be such that they include all parts of a particular object as a whole. While larger values of C result in a case that different parts of objects fall into different communities.

We define a community-driven data graph $G^{(C)} = (V^{(C)}, A^{(C)})$ where $V^{(C)}$, $A^{(C)}$ represent the nodes and edges of this graph, respectively. Each node in graph $G^{(C)}$ represents a detected community in graph $G^{(I)}$. $A_{ij}^{(C)}$ repre-

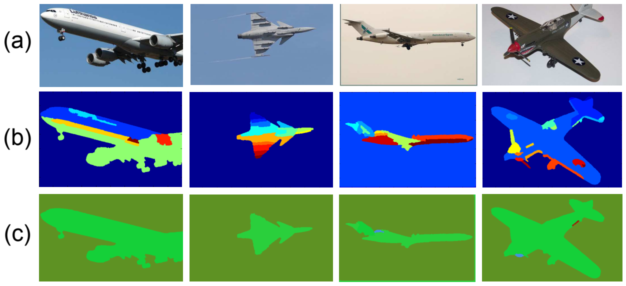

Figure 3: Row (a) illustrates four sample images from the VOC 2011 database. Row (b) represents the corresponding segmentations of the samples images. Each color denotes a segmented region. The segmentation is derived by the software provided by [13]. In row (c), each color denotes a community that the segmented regions of figures in row (b) belong to. These communities belong to the set of all detected communities over the entire database. The community detection algorithm is described in Section 2.2.

sents the number of links between the nodes of community i and community j in graph $G^{(I)}$.

2.3. Label Graph

Let $G^{(L)} = (V^{(L)}, A^{(L)})$ represent a label graph with $V^{(L)}$ and $A^{(L)}$ denoting the nodes and edges. Here, $V^{(L)}$ is a set of all labels presented in the database and $A^{(L)}_{ij}$ denotes the label correlation among two nodes i and j. Label correlation is determined using a set of training instances annotated for the label set. A binary vector v_i of length equal to the number of training instances is created for each label i. $v_i(j)$ is equal to one iff the jth training instance is annotated with the ith label. We use the standard cosine similarity to specify the correlation between the labels.

Since the database images may only be partially labeled by a subset of objects, we add an extra label called "background" for regions with no label associated to them in the database.

2.4. Initialization of Community Labels

In the fully labeled dataset where the background class is absent, each segmented region is initially associated with all of its image-level labels. We define initial labeling for each of the detected communities by matrix $Y = (Y_1, \dots, Y_{\mathcal{C}})'$ of dimension $\mathcal{C} \times \mathcal{K}$ with $\mathcal{C}$ being equal to the total number of detected communities and $\mathcal{K}$ denoting the total number of labels in the dataset. Y_{ij} represents the number of nodes in community c_i that are associated with label j. Multiple nodes from the same image in community c_i are counted as one. Each row vector Y_i, $i = \{1, \dots, \mathcal{C}\}$, is ℓ_1 normalized to give comparable label association to each of the communities independent of the number of nodes they contain. Next, we binarize each vector Y_i using a threshold β as:

$$Y_{ij} = \mathcal{I}(Y_{ij} > \beta). \tag{3}$$

For partially labeled datasets, we consider to have an additional column for Y corresponding to the background label. If $Y_{ij} = 0, \quad \forall j \in \{1, \dots, \mathcal{K}\}$, we set $Y_{i(\mathcal{K}+1)} = 1$ to associate that community with a "background" label. This is to compensate for partially labeled datasets. In the remainder of this paper, for partially labeled datasets, we assume $\mathcal{K}$ denotes the total number of labels in the dataset as well as the "background" label.

Algorithm 1 Mapping semantics to detected communities

```
**Input:** $G^{(I)}, A^{(L)}, A^{(C)}, \mathcal{C}, \mathcal{K}, \beta, c_i \quad \forall i \in \{1, \dots, \mathcal{C}\}$
**Output:** $U$
$R_{ij}$: set of nodes in community $c_i$ & image-level label $j$
**Comment:** Initialization of detected communities
$Y \leftarrow \mathcal{C} \times \mathcal{K}$ zero vector
for $i = 1 \rightarrow \mathcal{C}$ do
  for $j = 1 \rightarrow \mathcal{K}$ do
    if $|R_{ij}| \geq \beta$ then
      $Y(i, j) \leftarrow 1$
    end if
  end for
end for
**Comment:** Associate each community with a label
Solve Equation (9) for $X$
for $i = 1 \rightarrow \mathcal{C}$ do
  $U_{c_i} \leftarrow max(X(i, :))$
end for
```

2.5. Mapping Labels to Communities

We assume that every image segment represents a specific image label. By extending this idea, we want to associate each community with the most appropriate semantic label. In this section, we describe how each community is associated with a semantic label as shown in Figure 4. Let $A^{(C)}$ be an $\mathcal{C} \times \mathcal{C}$ affinity matrix of the community-driven graph with $\mathcal{C} = |V^{(C)}|$. Let $A^{(L)}$ be a $\mathcal{K} \times \mathcal{K}$ affinity matrix denoting the label graph constructed for the $\mathcal{K}$ concepts (labels). Let $X = (X_1, \dots, X_{\mathcal{C}})' = (E_1, \dots, E_{\mathcal{K}})$ be a $\mathcal{C} \times \mathcal{K}$ matrix defining the final labeling associated to every community derived in the previous section. Similarly, let $Y = (Y_1, \dots, Y_{\mathcal{C}})'$ be a $\mathcal{C} \times \mathcal{K}$ matrix denoting the initial label assignments to every community.

To assign a label to each community, we construct an optimization problem with the following properties:

(a) highly correlated labels to be assigned to highly correlated communities

(b) weakly connected communities have distinct labels

(c) small label deviation from initial label for each community.

Given a set of communities $\mathbb{C} = \{c_1, c_2, \dots, c_{\mathcal{C}}\}$, and their affinity matrix $A^{(C)}$, the objective function to classify each community with a unique label $\mathbb{L} = \{l_1, l_2, ..., l_K\}$ is defined as follows:

$$\Omega(X) = \underbrace{\frac{1}{2}\sum_{i,j=1}^{\mathcal{K}} A_{ij}^{(L)} \left\| \frac{E_i}{\sqrt{D_{ii}^{(L)}}} - \frac{E_j}{\sqrt{D_{jj}^{(L)}}} \right\|^2}_{(a)} + \underbrace{\lambda \sum_{i,j}^{\mathcal{C}} A_{ij}^{(C)} \|X_i - X_j\|^2}_{(b)} + \underbrace{\mu \sum_{i}^{\mathcal{C}} \|X_i - Y_i\|^2}_{(c)} . \quad (4)$$

where $D^{(L)}$ is a diagonal matrix whose (i, i) entries are equal to the sum of the i-th row of $A^{(L)}$, i.e. $D_{ii}^{(L)} = \sum_{j=1}^{\mathcal{K}} A_{ij}^{(L)}$. Next, the solution X that minimizes (4) is derived.

The first term on the right hand side of Equation (4) can be written as:

$$\begin{aligned} &\tfrac{1}{2} \sum_{i,j=1}^{\mathcal{K}} A_{ij}^{(L)} \left\| \tfrac{E_i}{\sqrt{D_{ii}^{(L)}}} - \tfrac{E_j}{\sqrt{D_{jj}^{(L)}}} \right\|^2 \\ &= \tfrac{1}{2} \left(\sum_{i=1}^{\mathcal{K}} E_i^T E_i + \sum_{j=1}^{\mathcal{K}} E_j^T E_j - 2 \sum_{i,j=1}^{\mathcal{K}} A_{ij}^{(L)} \tfrac{E_i^T E_i}{\sqrt{D_i^{(L)} D_j^{(L)}}} \right) \\ &= \sum_{i=1}^{\mathcal{K}} E_i^T E_i - \sum_{i,j=1}^{\mathcal{K}} A_{ij}^{(L)} \tfrac{E_i^T E_i}{\sqrt{D_i^{(L)} D_j^{(L)}}} \\ &= tr\left(E^T \left(I - {D^{(L)}}^{(-1/2)} A^{(L)} {D^{(L)}}^{(-1/2)} \right) E \right) \\ &= tr\left(X \mathbf{L}_1 X^T \right), \end{aligned} \quad (5)$$

with $\mathbf{L}_1 = D^{(L)} - {D^{(L)}}^{(-1/2)} A^{(L)} {D^{(L)}}^{(-1/2)}$. The second term in (4) can be written as:

$$\begin{aligned} &\sum_{i,j}^{\mathcal{C}} A_{ij}^{(C)} \|X_i - X_j\|^2 \\ &= \sum_{i}^{\mathcal{C}} \sum_{j}^{\mathcal{C}} A_{ij}^{(C)} (X_i - X_j)^T (X_i - X_j) \\ &= \sum_{i}^{\mathcal{C}} \sum_{j}^{\mathcal{C}} A_{ij}^{(C)} \left(X_i^T X_i + X_j^T X_j - 2X_j^T X_i \right) \\ &= 2 \sum_{i} X_i X_i^T \left(A_{ij}^{(C)} \right) - 2 \sum_{i} \sum_{j} A_{ij}^{(C)} X_i^T X_i \\ &= tr\left(X^T D^{(C)} X \right) - tr\left(X^T A^{(C)} X \right) \\ &= tr\left(X^T (D^{(C)} - A^{(C)}) X \right) \\ &= tr\left(X^T \mathbf{L}_2 X \right), \end{aligned} \quad (6)$$

with $\mathbf{L}_2 = D^{(C)} - A^{(C)}$. Thus, the cost function in (4) can be summarized as:

$$\Omega(X) = tr\left(X \mathbf{L}_1 X^T \right) + \lambda tr\left(X^T \mathbf{L}_2 X \right) + \mu \|X - Y\|^2. \quad (7)$$

$\Omega(X)$ in (7) is a convex function of X. By taking the derivative of (7) with respect to X, we get:

$$\frac{d(\Omega(X))}{dX} = 0 \rightarrow 2X\mathbf{L}_1 + 2\lambda \mathbf{L}_2 X + 2\mu(X - Y) = 0, \quad (8)$$

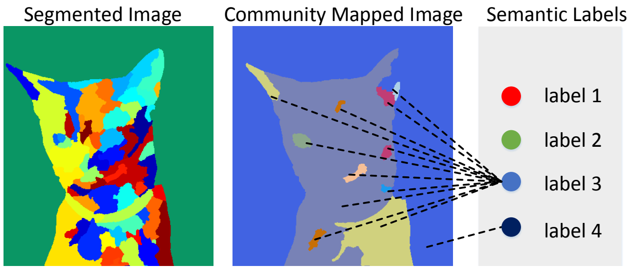

Figure 4: Illustration of mapping the communities to semantic labels. The colored circles on the right are denoting the semnatic labels across the dataset. Image is best viewed in color.

which can be written as:

$$(\mu I + \lambda \mathbf{L}_2) X + X \mathbf{L}_1 = \mu Y. \quad (9)$$

Equation (9) is a Lyapunov equation of the form $AQ + QB = C$. The solution to (9) can be obtained using existing software libraries.

Once X is found, we use the following formulation to assign a single label to each community c_i:

$$U_{c_i} = \arg\max_j \quad X_{ij}, \qquad \forall j \in \{1, ..., \mathcal{K}\}. \quad (10)$$

To maximize the performance, one can apply a line-search by stepping through the threshold β (threshold used to binarize the initial label matrix Y), and choose the one that results in the highest performance. This process is illustrated in Algorithm 1.

It is worth noting that instead of choosing the label with maximum likelihood for each community, one could add an additional constraint by defining each X_i as a binary vector and enforcing unit ℓ_1 norm of X_i (minimizing $\|X_i\|_1$). However, we did not follow such a formulation as it would require solving an integer programming problem which results in a non-convex optimization.

Figure 2 illustrates the image-driven graph, community-driven graph, label graph, along with the initial and final label estimations for each of the detected communities.

2.6. Generalizing to a Test Image

First, we segment each test image Q. To determine the association between each of its segmented regions and the detected communities, we follow [37] and provide a brief summary. Let $\mathcal{H}_i$ denote the set of all nodes in the spatial neighborhood of node i, c_j be a community with $j \in \{1, \ldots, \mathcal{C}\}$, and $\mathcal{T'}_i$ denote the set of all nodes that are in the top T' nearest neighbors of node i. The strength of association of a node i to a community c_j is measured by two factors: first by the attribute similarity between node i and community c_j, second by considering the attribute similarity between neighbors of node i and different communities in the network along with the relation between community c_j and each of the communities in the network.

Let $g(i \in c_j)$ denote the attribute similarity between node i and community c_j. The function $g(i \in c_j)$ is defined by the fraction of top T' nearest neighbors to node i that belong to community c_j:

$$g(i \in c_j) = \frac{\sum\limits_{k \in \mathcal{T}'_k} \mathcal{I}\{k \in c_j\}}{T'}. \tag{11}$$

Moreover, $f(c_{j'}, c_j)$ is defined to measure the relation between two communities $c_{j'}$ and c_j:

$$f(c_{j'}, c_j) = \frac{\sum\limits_{i \in c_{j'}} \sum\limits_{k \in c_j} \mathcal{I}\left\{A_{ik}^{(I)} > 0\right\}}{\sum\limits_{i \in c_{j'}} \sum\limits_{k=1}^{|V^{(I)}|} \mathcal{I}\left\{A_{ik}^{(I)} > 0\right\}} \tag{12}$$

where $|V^{(I)}|$ denotes the total number of nodes in the image-driven graph. In particular, $f(c_{j'}, c_j)$ measures the number of links between the two communities $c_{j'}$ and c_j divided by the total number of links between community $c_{j'}$ and all other communities. Thus, the strength of association of a node i to a community c_j can be determined by $\pi_j^{(i)}$:

$$\pi_j^{(i)} = \frac{\sum\limits_{k \in \mathcal{H}_i} \left[\sum\limits_{j'=1}^{\mathcal{C}} f(c_{j'}, c_j) g(k \in c_{j'}) \right] g(i \in c_j)}{\sum\limits_{j''=1}^{\mathcal{C}} \sum\limits_{k \in \mathcal{H}_i} \left[\sum\limits_{j'=1}^{\mathcal{C}} f(c_{j''}, c_{j'}) g(k \in c_{j'}) \right] g(i \in c_{j''})}, \tag{13}$$

where $\mathcal{C}$ denotes the total number of detected communities. Finally, each segmented region i is mapped to community $\hat{c}$ with the highest association:

$$\hat{c} = \arg\max_{j} \quad \pi_j^{(i)}, \tag{14}$$

and is semantically labeled accordingly.

3. Experimental Results

To demonstrate the applicability of the proposed method, four commonly used datasets for semantic segmentation are considered: MSRC-21 [12], LabelMe [28], VOC 2009 [15] and VOC 2011 [16]. Our approach does not require groundtruth segmentation in the training phase.

To measure the performance, we choose the common average per-class accuracy which measures the percentage of correctly classified pixels for a class and then averages over all classes. This measurement avoids bias towards classes that occupy larger image areas, such as sky or grass. It also penalizes a model that maximizes agreement simply by predicting very few labels overall.

We investigate how the performance of our method varies as the level of segmentation changes. We further evaluate the performance of our system as a function of the number of detected communities $\mathcal{C}$. In addition, the sensitivity of our approach on parameter β, the threshold used to binarize initial label vectors, is illustrated. Furthermore, the performance of this work is compared against the state of the art weakly supervised semantic segmentation methods as well as fully supervised methods. Finally, the computational cost of the proposed approach is discussed.

3.1. Database

In our experiments, we use MSRC-21, LabelMe, and VOC 2009 and VOC 2011 datasets. We do not use any location information associated with the label annotations in training or testing phase for any of these datasets. At training phase, only the image-level labels are considered.

MSRC-21: This is a publicly available dataset containing 591 images. This dataset is a fully annotated dataset with 21 object classes. To make a fair comparison, we evaluate the performance of our system on MSRC-21 as the majority of the weakly supervised approaches are also evaluated on this dataset. Ground-truth semantic segmentation information of MSRC-21 is only used to measure the accuracy of the proposed system.

LabelMe: This is a more challenging dataset than MSRC-21. It includes $2,688$ images from 33 classes and is fully annotated. Similar to MSRC-21 dataset, the groundtruth semantic segmentation information is only used to measure the accuracy of our system.

VOC 2009, 2011: This dataset is a publicly available dataset with partial labeling. There are 20 labeled objects and 1 background class. We choose to evaluate the performance on VOC as each image contains multiple classes while being only partially labeled. This is a more challenging dataset as a single background class covers several object classes such as “sky” and “grass”. It addition, there are more significant background clutter, illumination changes and occlusions.

3.2. Evaluation

Performance: In Figure 5, we demonstrate the performance when the segmentation level varies. By setting the parameters of [13], we achieve three levels of segmentation with an average number of 20, 50, and 70 segments per image which are referred to as “Coarse”, “Mid”, and “Fine”, respectively. It can be seen that “Fine” segmentation level has the highest performance accuracy. This can be predicted as at “Fine” level more spatial information is encoded which provides different objects with more accurate outlines and thus higher performance. Furthermore, Figure 5 illustrates the effect of the number of detected communities on the performance accuracy. For MSRC-21 dataset, the performance of the proposed approach remains nearly invariant when the number of detected communities is larger than 500.

Figure 6 illustrates the performance as a function of threshold β. As expected, extremely large or small values of β can degrade the performance. This effect can be justified as large values result in removing the majority of

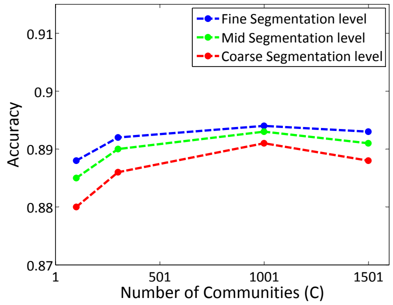

Figure 5: Effect of different levels of segmentation as well as different number of detected communities on the performance accuracy. Results are reported for MSRC-21 dataset.

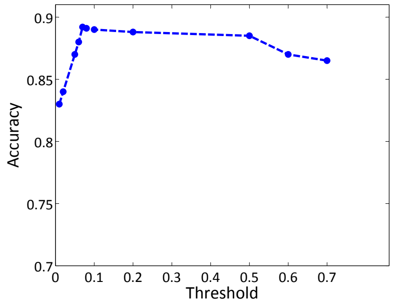

Figure 6: Performance of the proposed appraoch against threshold β. Results are reported for MSRC-21 dataset.

initial community labelings while small β results in having a large group of initial labelings per image. We notice that our method is robust to selection of the parameter β as well.

The per-class performance accuracy between the proposed approach and the baseline methods [44], [48], and [53] is compared in Table 1. Our approach achieves a higher average per-class accuracy than the baseline methods. We have also included the state of the art results of fully-supervised methods [7], [3], [36], and [50] reported on MSRC-21 dataset. Our approach is even comparable with the fully-supervised ones without requiring ground-truth labeling at training time.

Furthermore, Table 2 demonstrate that the average per-class accuracy of the proposed work on LabelMe dataset compares competitively with the state of the art results. The approach of [31] achieves an average accuracy of 51.7% while being fully supervised (requires ground truth segmentation in the training phase) and involves training a convolutional neural network (CNN). Again, we note that our approach does not require complete supervision. In general, the learning capacity of these networks depends on the number and size of the kernels in each layer and the number of kernel combinations between layers. The work of [31] has roughly more than a million parameters which leads to the training time of more than three days on a GPU. A detailed analysis of computational complexity of our approach shall be discussed shortly.

Database: MSRC-21								
Method	[7]	[3]	[36]	[50]	[44]	[48]	[53]	ours
	FS	FS	FS	FS	WS	WS	WS	WS
building	_	66	70	**71**	45	12	_	**89**
grass	_	87	**98**	**98**	64	83	_	**97**
tree	_	84	87	**90**	71	70	_	**89**
cow	_	**81**	76	79	75	81	_	**94**
sheep	_	83	79	**86**	74	**93**	_	92
sky	_	93	**96**	93	86	84	_	**96**
airplane	_	81	81	**88**	81	**91**	_	89
water	_	82	75	**86**	47	55	_	**87**
face	_	78	86	**90**	1	**97**	_	88
car	_	**86**	74	84	73	87	_	**96**
bicycle	_	**94**	88	**94**	55	**92**	_	89
flower	_	96	96	**98**	**88**	82	_	87
sign	_	**87**	72	76	6	69	_	**90**
bird	_	48	36	**53**	6	51	_	**82**
book	_	90	90	**97**	63	61	_	**89**
chair	_	**81**	79	71	18	59	_	**79**
road	_	82	87	**89**	**80**	66	_	77
cat	_	82	74	**83**	27	53	_	**87**
dog	_	**75**	60	55	26	44	_	**89**
body	_	**70**	54	68	55	9	_	**88**
boat	_	**52**	35	17	8	58	_	**96**
average	**94.5**	80	76	79.3	50	67	80	**89**

Table 1: Per-class accuracy on MSRC-21 using the proposed approach, state of the art fully supervised approaches ([7], [3], [36], [50]) and weakly supervised methods ([44], [48], [53]). FS and WS denote fully-supervised and weakly-supervised approaches. The best results achieved by FS and WS approaches are highlighted in bold. Our method achieves 9% improvement compared to WS approaches, while being comparable to FS approaches.

Database: LabelMe							
Method	[41]	[43]	[31]	[44]	[30]	[53]	ours
	FS	FS	FS	WS	WS	WS	WS
Accuracy	13	41	**51.7**	14	25	33.37	**44**

Table 2: Performance accuracy on LabelMe dataset using the proposed approach, state of the art fully supervised approaches ([41], [43], [31]) and weakly supervised methods ([44], [30], [53]). The best results achieved by FS and WS approaches are highlighted in bold. We achieve 11% improvement compared to WS approaches, while being comparable to FS approaches.

By applying the proposed approach to partially labeled VOC 2011 dataset, we achieve an average accuracy of 43.2%. To the best of our knowledge, no weakly supervised results were reported on this dataset yet. Table 3 reports the results achieved by our approach and the state of the art fully-supervised methods. It can be seen that our approach is achieving comparable performance while being only weakly supervised. It is worth noting that [20] achieves its performance using a pre-trained CNN on a large auxiliary dataset and then fine-tuned for VOC 2011. Also, [5] uses additional external ground truth segmentations.

Table 4 illustrates that the proposed approach achieves higher performance accuracy than the state of the art weakly supervised approaches which are trained and tested on VOC 2009 dataset.

In addition, Figure 7 illustrates step-by-step qualitative results on the fully labeled MSRC-21 dataset and the partially labeled VOC 2011 dataset. In this Figure, label "void"

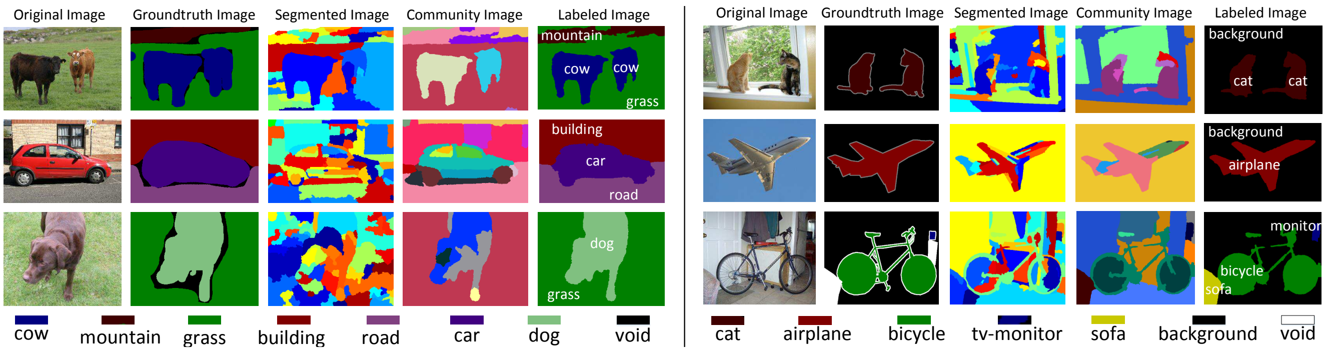

Figure 7: Qualitative results of the fully labeled MSRC-21 (left) and the partially labeled VOC 2011 (right) datasets. Colors do not represent the same concept across the columns and across datasets. Color labeling at the bottom denotes object classes of labled images.

Database: VOC 2011					
Method	[22] FS	[6] FS	[5] FS	[20] FS	ours WS
Accuracy	41.4	43.3	46.4	**47.9**	**43.2**

Table 3: Performance accuracy on VOC 2011 dataset using the proposed approach and state of the art fully supervised approaches [22], [6], [5] and [20]. The best results achieved by FS and WS approaches are highlighted in bold.

Database: VOC 2009				
Method	[48] WS	[1] WS	[53] WS	ours WS
Accuracy	38.3	39.2	47.5	**52.1**

Table 4: Performance accuracy on VOC 2009 dataset using the proposed approach and state of the art weakly supervised approaches ([48], [1], [53]). The best result achieved is highlighted in bold.

stands for pixels in the image that are ignored in evaluation for the particular dataset. As shown, the final labeled image follows the groundtruth image closely.

Scalability: The most computationally expensive part of our training method is to find the nearest neighbors to each node as part of the algorithm for constructing the image-driven graph. The simplest solution is to compute the distance from each node to all the other nodes in the graph. The computational cost of creating the image-driven graph can be significantly reduced by using a space partitioning data structure (k-d tree) which has a construction cost of $O(N \log N)$ and search cost of $O(\log N)$ for every node. The normalized cut based community detection algorithm is of complexity $O(N)$ (refer to [39]). Finally, the convex optimization algorithm to map semantic labels to the communities only depends on the number of communities which in practice varies linearly with the number of labels. The proposed approach takes roughly about 30 seconds to train on MSRC-21 dataset and 0.02 seconds per image at test time, which is significantly lower than the reported time of 7 seconds per image by the semi-supervised approach of [48] on computers with similar processing capabilities. Table 5 illustrates the comparison between training time and test time between our method and the approaches of [50], [36], and [48]. We speculate that the majority of the running time at test time in our method is reduced by only requiring to map each segment of the test image to one of the previously semantically labeled detected communities.

Method	[36] FS	[50] FS	[48] WS	ours WS
CPU Spec.	2.7 GHz 64-bit	3.4 GHz 64-bit	2.7 GHz 64-bit	3.0 GHz 64-bit
Total training time (sec)	800	12600	—	**30**
Test time (sec/ image)	1	7.3	7	**0.02**

Table 5: Comparison of computational time on MSRC-21 dataset for the proposed approach with state of the art fully supervised approaches ([36], [50]) and baseline weakly supervised method of [48]. The proposed work is more than 300 times faster than its baseline weakly supervised method.

4. Conclusion

We presented a graph based weakly supervised semantic segmentation by learning communities of image-parts. Pixel-level annotations for a given test image is provided through mapping its segmented regions to each of the learned semantically labeled communities. Extensive experiments conducted on challenging datasets demonstrate that the proposed approach compares favorable with current state of the art methods. In addition, it is shown that our work is considerably more computationally efficient than the baseline approaches. Therefore, one can easily scale up to large datasets. Our future work focuses on ways of solving the problem of bottom-up and top-down segmentations jointly. Another potential research direction would be to incrementally update communities as more images are provided. Due to the computational efficiency of our method, in future we want to explore the applicability of the proposed method on semantic segmentation of videos.

5. Acknowledgment

This work is supported by ONR grant #N00014-12-1-0503 and UCSB Doctoral Scholar Fellowship.

References

[1] A. Vezhnevets, V. Ferrari, and J. M. Buhmann. Weakly supervised structured output learning for semantic segmentation. In *CVPR*, 2012.

[2] P. Arbeláez, B. Hariharan, C. Gu, S. Gupta, L. Bourdev, and J. Malik. Semantic segmentation using regions and parts. In *CVPR*, 2012.

[3] X. Boix, J. M. Gonfaus, J. van de Weijer, A. D. Bagdanov, J. Serrat, and J. Gonzàlez. Harmony potentials. *IJCV*, 2012.

[4] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2009.

[5] J. Carreira, R. Caseiro, J. Batista, and C. Sminchisescu. Semantic segmentation with second-order pooling. In *ECCV*. 2012.

[6] J. Carreira, F. Li, and C. Sminchisescu. Object recognition by sequential figure-ground ranking. *IJCV*, 2012.

[7] J. Carreira and C. Sminchisescu. Cpmc: Automatic object segmentation using constrained parametric min-cuts. *PAMI*, 2012.

[8] F.-J. Chang, Y.-Y. Lin, and K.-J. Hsu. Multiple structured-instance learning for semantic segmentation with uncertain training data. In *CVPR*, 2014.

[9] L.-C. Chen, G. Papandreou, and A. L. Yuille. Learning a dictionary of shape epitomes with applications to image labeling. In *ICCV*, 2013.

[10] X. Chen, A. Jain, A. Gupta, and L. S. Davis. Piecing together the segmentation jigsaw using context. In *CVPR*, 2011.

[11] X. Chen, A. Shrivastava, and A. Gupta. Enriching visual knowledge bases via object discovery and segmentation. In *CVPR*, 2014.

[12] A. Criminisi, T. Minka, and J. Winn. Microsoft research cambridge object recognition image dataset. version 2.0, 2004. Available at *http://research.microsoft.com/en-us/projects/objectclassrecognition/*.

[13] Y. Deng and B. Manjunath. Unsupervised segmentation of color-texture regions in images and video. *PAMI*, 2001.

[14] J. Dong, Q. Chen, S. Yan, and A. Yuille. Towards unified object detection and semantic segmentation. In *ECCV*. 2014.

[15] M. Everingham, L. Van Gool, C. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge 2009. In *2th PASCAL Challenge Workshop*, 2009.

[16] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 2010.

[17] C. Farabet, C. Couprie, L. Najman, and Y. LeCun. Learning hierarchical features for scene labeling. *PAMI*, 2013.

[18] B. Fulkerson, A. Vedaldi, and S. Soatto. Class segmentation and object localization with superpixel neighborhoods. In *CVPR*, 2009.

[19] C. Galleguillos, B. Babenko, A. Rabinovich, and S. Belongie. Weakly supervised object localization with stable segmentations. In *ECCV*, 2008.

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

[21] A. Gupta, A. A. Efros, and M. Hebert. Blocks world revisited: Image understanding using qualitative geometry and mechanics. In *ECCV*, 2010.

[22] A. Ion, J. Carreira, and C. Sminchisescu. Probabilistic joint image segmentation and labeling. In *Advances in Neural Information Processing Systems*, 2011.

[23] P. Kohli, P. H. Torr, et al. Robust higher order potentials for enforcing label consistency. *IJCV*, 2009.

[24] M. P. Kumar and D. Koller. Efficiently selecting regions for scene understanding. In *CVPR*, 2010.

[25] L. Ladicky, C. Russell, P. Kohli, and P. H. Torr. Associative hierarchical crfs for object class image segmentation. In *CVPR*, 2009.

[26] L. Ladicky, C. Russell, P. Kohli, and P. H. Torr. Graph cut based inference with co-occurrence statistics. In *ECCV*, 2010.

[27] C. Li, D. Parikh, and T. Chen. Automatic discovery of groups of objects for scene understanding. In *CVPR*, 2012.

[28] C. Liu, J. Yuen, and A. Torralba. Nonparametric scene parsing: Label transfer via dense scene alignment. In *CVPR*, 2009.

[29] C. Liu, J. Yuen, and A. Torralba. Nonparametric scene parsing via label transfer. *PAMI*, 2011.

[30] Y. Liu, J. Liu, Z. Li, J. Tang, and H. Lu. Weakly-supervised dual clustering for image semantic segmentation. In *CVPR*, 2013.

[31] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. *arXiv preprint arXiv:1411.4038*, 2014.

[32] D. Lowe. Object recognition from local scale-invariant features. In *ICCV*, 1999.

[33] T. Malisiewicz and A. Efros. Beyond categories: The visual memex model for reasoning about object relationships. In *NIPS*, 2009.

[34] T. Malisiewicz and A. A. Efros. Recognition by association via learning per-exemplar distances. In *CVPR*, 2008.

[35] M. S. NIKULIN. *Hellinger Distance*. Springer, 2001.

[36] D. Pei, Z. Li, R. Ji, and F. Sun. Efficient semantic image segmentation with multi-class ranking prior. *Computer Vision and Image Understanding*, 2014.

[37] N. Pourian and B. Manjunath. Pixnet: A localized feature representation for classification and visual search. *Trans. on Multimedia*, 2015.

[38] M. Rubinstein, A. Joulin, J. Kopf, and C. Liu. Unsupervised joint object discovery and segmentation in internet images. In *CVPR*, 2013.

[39] J. Shi and J. Malik. Normalized cuts and image segmentation. *PAMI*, 2000.

[40] J. Shotton, M. Johnson, and R. Cipolla. Semantic texton forests for image categorization and segmentation. In *CVPR*, 2008.

[41] J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *ECCV*, 2006.

[42] J. Tighe and S. Lazebnik. Superparsing: scalable nonparametric image parsing with superpixels. In *ECCV*, 2010.

[43] J. Tighe and S. Lazebnik. Finding things: Image parsing with regions and per-exemplar detectors. In *CVPR*, 2013.

[44] J. Verbeek and B. Triggs. Region classification with markov field aspect models. In *CVPR*, 2007.

[45] J. Verbeek and W. Triggs. Scene segmentation with crfs learned from partially labeled images. In *NIPS*, 2008.

[46] A. Vezhnevets and J. M. Buhmann. Towards weakly supervised semantic segmentation by means of multiple instance and multitask learning. In *CVPR*, 2010.

[47] A. Vezhnevets, J. M. Buhmann, and V. Ferrari. Active learning for semantic segmentation with expected change. In *CVPR*, 2012.

[48] A. Vezhnevets, V. Ferrari, and J. M. Buhmann. Weakly supervised semantic segmentation with a multi-image model. In *ICCV*, 2011.

[49] L. Yang, P. Meer, and D. J. Foran. Multiple class segmentation using a unified framework over mean-shift patches. In *CVPR*, 2007.

[50] J. Yao, S. Fidler, and R. Urtasun. Describing the scene as a whole: Joint object detection, scene classification and semantic segmentation. In *CVPR*, 2012.

[51] H. Zhang, J. E. Fritts, and S. A. Goldman. Image segmentation evaluation: A survey of unsupervised methods. *Computer Vision and Image Understanding*, 2008.

[52] L. Zhang, M. Song, Z. Liu, X. Liu, J. Bu, and C. Chen. Probabilistic graphlet cut: Exploiting spatial structure cue for weakly supervised image segmentation. In *CVPR*, 2013.

[53] L. Zhang, Y. Yang, Y. Gao, Y. Yu, C. Wang, and X. Li. A probabilistic associative model for segmenting weakly-supervised images. *Trans. Image Processing*, 2014.

[54] T. Zhou, Y. Jae Lee, S. X. Yu, and A. A. Efros. Flowweb: Joint image set alignment by weaving consistent, pixel-wise correspondences. In *CVPR*, 2015.

[55] L. Zhu, Y. Chen, Y. Lin, C. Lin, and A. Yuille. Recursive segmentation and recognition templates for image parsing. *PAMI*, 2012.