Efficient Scene Labeling via Sparse Annotations

Can Qin[1,*] **Maoguo Gong**[1] **Yue Wu**[2,*] **Dayong Tian**[3]
[1]Key Laboratory of Intelligent Perception and Image Understanding, Xidian University, Xi'an, China
[2]School of Computer Science and Technology, Xidian University, Xi'an, China
[3]School of Electronics and Information, Northwestern Polytechnical University, Xi'an, China
canqinn@gmail.com, gong@ieee.org, ywu@xidian.edu.cn, dayong.tian@nwpu.edu.cn
*Corresponding author

Abstract

Scene labeling (SL) in images is a significant part of Visual Internet of Things (VIoT). Most of existing approaches for SL employ fully supervised methods requiring massive pixel-wise annotations which are costly and time-consuming to obtain. To further deduce the required amount of manually labeled data, we propose a semi-supervised SL paradigm based on the joint optimization of deep representation and scene clustering. In order to learn the deep representation with semantic-friendly distribution, we design a novel constrained clustering which is composed of two steps: (1) overclustering deep features into raw clusters with high self-consistence; (2) introducing sparse annotations as semantic constrains to merge raw clusters into scene clusters. Experimental results show that the proposed approach has achieved satisfying performance on SIFT Flow and Stanford Background benchmarks by leveraging very few annotations (0.1% or less).

On the Internet of Things (IoT), enormous amounts of data acquired by different sensors should be stored, processed and presented in interpretable forms (Gubbi et al. 2013). Visual Internet of Things (VIoT), consisting of visual sensors, wireless transmission and intelligent analysis, is a critical yet challenging topic in IoT (Ma et al. 2016; Zhang, Wang, and Jia 2013). Most intelligent analysis, e.g. object detection or face recognition, requires a huge amount of manually labeled data (Zhang and Huang 2017). For example, scene recognition models trained with dense annotations enjoy the state-of-the-art accuracy (Bearman et al. 2016), yet obtaining such annotations manually is very time-consuming and costly. The annotation time of an indoor image containing 23 objects is tens of minutes (Bearman et al. 2016) which is a heavy burden for researchers. Hence, semi-supervised and weakly supervised Scene Labeling (SL) methods that parse a full image by labeling the semantic category of each pixel with a few manual labels attract wide interests.

Many semi or weakly supervised semantic segmentation methods which are similar to SL methods have been explored (Kwak, Hong, and Han 2017; Papandreou et al. 2015; Bearman et al. 2016). While the difference is that most of them are designed to annotate parts of objects within an image rather than labeling the full image. Generative Adversarial Network (GAN) has been applied to deal with semi-supervised SL tasks (Souly, Spampinato, and Shah 2017), but it requires too many annotations, 7% or more, for training. Recently, a new paradigm based on the combination of Convolutional Neural Network (CNN) and clustering has been explored. Yang *et al.* introduce CNN to optimize the features towards a clustering-friendly distribution so as to improve the performance of unsupervised clustering (Yang, Parikh, and Batra 2016). Hsu and Lin introduce a recurrent CNN to jointly solve clustering and representation learning in an iterative manner (Hsu and Lin 2017).

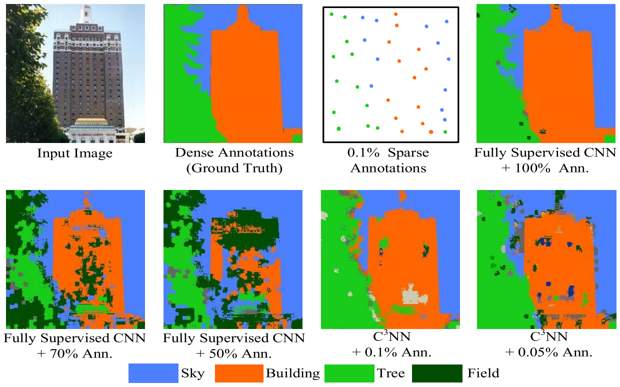

Input Image
Dense Annotations (Ground Truth)
0.1% Sparse Annotations
Fully Supervised CNN + 100% Ann.
Fully Supervised CNN + 70% Ann.
Fully Supervised CNN + 50% Ann.
C^3NN + 0.1% Ann.
C^3NN + 0.05% Ann.
Sky
Building
Tree
Field

Figure 1: Visual comparison of fully supervised CNN and C^3NN in leveraging different ratios of annotated labels. Ann. refers to annotations and '+ percentage' represents the ratio of annotations exploited for training. The sparse annotations could be manually or randomly selected from dense annotations. C^3NN outperforms fully supervised CNN by leveraging tens or hundreds of times less annotations.

Inspired by previous works, we design a new framework named Constrained Clustering Based Convolutional Neural Network (C^3NN) for semi-supervised SL. C^3NN involves a joint optimization of constrained clustering and CNN-based representation learning. A typical approach of constrained clustering imposes pairwise constraints between examples by specifying a limited number of *must-link* or *cannot-link* constraints (Wagstaff et al. 2001). The constrained clustering carried in C^{3}NN incorporates a new strategy, which is composed of two steps: over-clustering and cluster merging. Our approach achieves multiple tiny-sized raw clusters with high self-consistence by overly clustering the features. Next, sparse labeled data, served as semantic constraints, help recognizing and merging the raw clusters according to their semantic attributes. New scene clusters after processed by constrained clustering form a semantic-friendly distribution. CNN, fed by the results of constrained clustering, learns the hierarchical representations of semantic-friendly labels (Farabet et al. 2013). All the raw features are encoded into CNN-based features in high-level concepts representing their semantic attributes in a more abstract and precise way (Bengio, Courville, and Vincent 2013), which are more friendly for constrained clustering in the next episode. Through the cooperation between the constrained clustering and representation learning, discriminative scene clusters are distinguished by machines.

It is the first attempt to incorporate semi-supervised clustering with CNN for SL which is a smart way towards VIoT data analysis. A novel constrained clustering has been introduced, which could group diversified local features into the clusters with semantic-friendly distribution. Another innovation of C^{3}NN is designing a new cooperative paradigm between representation learning and constrained clustering. It has continuously refined semantic-friendly deep representation by jointly optimizing these two modules. C^{3}NN has reserved a high labeling accuracy by leveraging very few annotated labels, i.e. 0.1% or less. As shown in Figure 1, fully supervised CNN trained with partial annotations is inferior to C^{3}NN in a large scale, while the sparse annotations of an image require manually labeling only tens of pixels. It takes dozens of seconds for an annotator to achieve it which can easily meet the demands of manually labeling for the increasing amounts of VIoT data.

The rest of the paper is organized as follows. We first briefly introduce related works. Next, we present the proposed model and algorithm. Finally, the experimental results and analysis are given.

Related Works

Typical approaches of SL aim at exploiting graph structures within the image, applying Markov Random Field (MRF) or Conditional Random Field (CRF) to capture the local features so as to use them as the classifiers to annotate semantic categories of each pixel (Gould, Fulton, and Koller 2009; Triggs and Verbeek 2008). These methods are trained with hand craft features which are unpractical nowadays.

Recently, deep learning (DL) has made great advances in computer vision. CNN, regarded as an effective tool for feature extraction, has been widely applied in SL tasks (Pinheiro and Collobert 2014; Caesar and Ferrari 2016; Farabet et al. 2013). Chen *et al.* incorporate fully connected CRF with CNN (Chen et al. 2014), which makes a boost in accuracy. Currently, Fully Convolutional Networks (FCN) (Wang et al. 2017; Pan et al. 2017) plays an important role in SL. FCN could take images with any sizes and has shown significant improvements in performance. However, all these methods need dense labeled data which are time-consuming to obtain. The high cost of annotations makes smart solutions like semi or weekly supervised SL paradigms required.

Image-level labels (Kwak, Hong, and Han 2017; Papandreou et al. 2015) and bounding boxes (Papandreou et al. 2015) are commonly designed for weekly supervised semantic segmentation. However, these methods are not designed for labeling scenes in a full image and the accuracy of them is uncompetitive compared to those of fully supervised methods. GAN has been introduced to deal with semi-supervised SL tasks (Souly, Spampinato, and Shah 2017), while it depends on too many annotations.

Constrained Clustering Based Convolutional Neural Networks

As illustrated in Figure 2, C^{3}NN is a recurrent framework composed of two modules: constrained clustering and CNN-based representation learning. All the details are as follows.

CNN-based representation learning

The image set $I = \{I_1, I_2, ..., I_N\}$ with N images has been divided into M patches with the center of each superpixel by the SLIC algorithm (Achanta et al. 2012) where $M = N \times m$ and there are *m* patches within each image. Half of the patches are in the size of $w \times w \times 3$ and the others are in $w^{'} \times w^{'} \times 3$ where $w' > w$. The patches of larger size w' have been downsampled into the patches with size w by Gaussian pyramid (Lowe 2004) in order to feed CNN with multi-scaled inputs. The patches of the *n*-th image could be formulated as $Z_n = \{z_{i,n} \in \mathbb{R}^{\mathrm{V}} | i = 1, 2, ..., m; V = w \times w \times 6\}$ where $Z_n \subseteq Z$ and Z is defined as the full set of all the patches of the data set. As illustrated in Figure 2, X_n denotes the features extracted from the CNN of the *n*-th image, where $X_n \in X$ and X is defined as the collection of extracted features of the full set. X_n could be formulated as:

$$X_n = f\left(Z_n; \Theta\right) \tag{1}$$

where Θ denotes the parameters of CNN and f represents the forward pass function of CNN.

Constrained clustering

In this paper, we propose a novel scheme of constrained clustering which involves a combination of K-means clustering (Lloyd 1982) and cluster merging. It firstly achieves multiple tiny-sized raw clusters with high self-consistence by overclustering the features. In order to isolate the cross-image interference, we divide the global clustering space into N subsets according to their original images so that the features of each image are processed independently. Then, apply K-means clustering to group data points $X^n = \{\mathbf{x}_{i,n} \in \mathbb{R}^p | i = 1, 2, ..., m\}$ into K raw clusters where p is the dimensionality of $x_{i,n}$. K-means clustering separates the features by minimizing the following function:

$$\min_{M_n \in \mathbb{R}^{p \times K}, \mathbf{s}_{i,n} \in \mathbb{R}^K} ||\mathbf{x}_{i,n} - M_n \mathbf{s}_{i,n}||_2^2 \tag{2}$$
$$s.t. \quad s_{j,i,n} \in \{0, 1\}, \mathbf{1}^T \mathbf{s}_{i,n} = 1, \forall i, j.$$

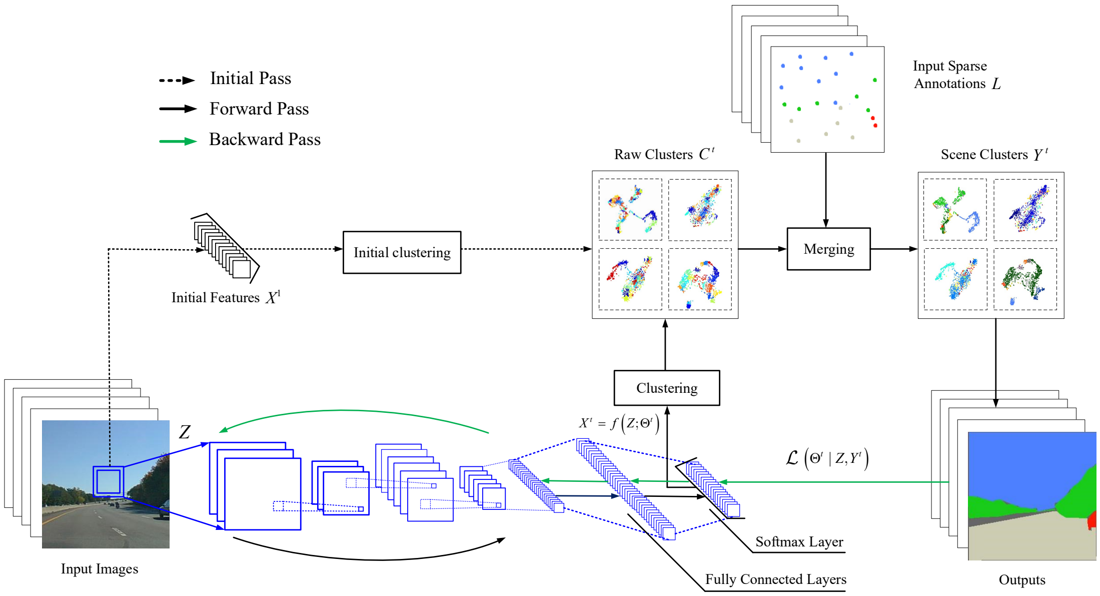


Figure 2: Learning scheme of Constrained Clustering Based Convolutional Neural Networks (C^{3}NN). C^{3}NN begins with the initial pass and then, it would be carried in backward and forward pass iteratively until meeting the desired number of clusters.

where $\mathbf{s}_{i,n}$ is the assignment vector of data point $\mathbf{x}_{i,n}$. $s_{i,j,n}$ represents the j-th element of $\mathbf{s}_{i,n}$ and the k-th column of M_n denotes the centroid of the k-th cluster. Given the set S, the full collection of raw clusters' index vectors, where $\mathbf{s}_{i,n} \in S_n \subseteq S$, the assignments of all clusters C are:

$$C = \{C_n\} = \sum_{i=1}^{m} i \cdot \mathbf{s}_{i,n}, n = 1, ...N \qquad (3)$$

where m means the quantity of patches per image. We denote the clusters attached with same assignments by $C_n = \{c_{k,n}\}, k = 1, ..., m_k$ where m_k represents the quantity of clusters within C_n.

Overclustering help grouping features to multiple tiny-sized raw clusters with high-consistence. However, SL requires organizing the features based on semantic attributes. Sparse labeled data, served as semantic constraints, help recognizing the raw clusters by evaluating the semantic category that most of sparse annotations within this raw cluster belong to. The raw clusters belonging to the same category would be merged into one scene cluster. All the annotated labels within the image set are represented by a sparse matrix $L = [l_{i,j,n}]_{K\times M\times N}$ where $l_{i,j,n} \in \{0, 1\}$. $l_{i,j,n} = 1$ means that the annotated label of the i-th superpixel of the n-th image is j and $\mathbf{1}^T \mathbf{l}_{i,:,n} = 0$ represents that the i-th superpixel contains no annotated labels. The process of clusters merging can be formulated as:

$$y_n = \arg\max_{y_n \in \kappa} \sum_{k=1}^{m_k} L(y_n, c_{k,n}, n) \qquad (4)$$
$$s.t. \;\; \mathbf{1}^T \mathbf{l}_{i,:,n} \in \{0, 1\}, \forall i, n$$

where y_n represents the outputs of constrained clustering for the n-th image and m_k represents the quantity of clusters within C_n. $L(y_n, c_{k,n}, :)$ denotes the element at the y_n-th row and $c_{k,n}$-th column in the matrix. The assignment set of annotated labels is denoted by $\kappa = \{1, 2, ..., K\}$.

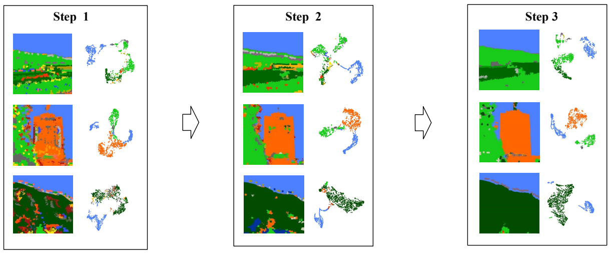


Figure 3: Performance of C^{3}NN and corresponding scene clusters through recurrent optimization. Both scene clusters and SL performance have been jointly improved by C^{3}NN.

The raw clusters without any sparse annotations would be unprocessed and assigned as the scene of 'unknown'. Since the scene clusters form a semantic-friendly distribution, raw clusters grouped by K-means clustering in the next episode could be expanded. Through the recurrent optimization, the quantity of raw clusters would decrease exponentially until meeting the desired number of clusters K^* suggested to be one or two times larger than the quantity of existed scenes in images.

Recurrent optimization

As illustrated in Figure 3, both the performance and distribution of scene clusters have been jointly optimized. In the framework, there are two groups of parameters need to be solved, i.e. $\{\Theta, Y\}$. The optimization can be implemented by carrying out in forward pass and backward pass.

Initialization and forward pass optimization The initial labels for training CNN are generated by the over-clustering features extracted from all the patches:

$$\{Z_n\} \to X^1, n = 1, ..., N \tag{5}$$

where $\mathbf{x}_n^1 \in X^1 \subseteq \mathbb{R}^{\mathrm{V}}$ and $V = w \times w \times 6$. $\to$ denotes the transformation from matrix set $\{Z_n\}$ into vector set X^1 by stacking the vectors of the matrixes in $\{Z_n\}$. X^1 is also applied as the initial features. In later episodes, we get features X^t collected from feature extraction layer of CNN with fixed parameters Θ^t where $\Theta^t \subseteq \Theta$. The sparse annotated labels denoted by the matrix L are required to regulate the outputs of K-means clustering. The optimization procedure in forward pass is formulated as:

$$\begin{cases} s_{i,n}^t = \underset{M^t, \mathbf{s}_{i,n}^t}{\arg\min} \sum\limits_{i=1}^{m} \left\| \mathbf{x}_{i,n}^t - M^t \mathbf{s}_{i,n}^t \right\| \\ c_n^t = \sum\limits_{i=1}^{m} i \cdot \mathbf{s}_{i,n}^t \\ y_n^t = \underset{y_n^t \in \kappa}{\arg\max} \sum\limits_{k=1}^{m_k} L(y_n^t, c_{k,n}^t, n) \end{cases} \tag{6}$$

where superscript t represents the t-th episode. The output set of all images is Y^t where $y_n^t \in Y^t \subseteq Y$.

Backward pass optimization In backward pass, it is a standard supervised representation learning procedure based on the output labels generated by constrained clustering.

$$\mathcal{L}(\Theta^t | Z, Y^t) = \sum_{i=1}^{N} \left\| f\left(Z_i; \Theta^t\right) - y_i^t \right\| \tag{7}$$

The parameters of CNN would be updated by optimizing the previous loss function:

$$\Theta^t = \underset{\Theta^t}{\mathrm{argmin}} \mathcal{L}(\Theta^t | Z, Y^t) \tag{8}$$

The loss function $\mathcal{L}(\Theta^t | Z, Y^t)$ is optimized by the backpropagation algorithm (Rumelhart and Mcclelland 1986).

Experiments

Datasets and measurements

As the C^{3}NN aims at performing a full image SL output, we select two benchmarks to justify our method: SIFT Flow (Liu, Yuen, and Torralba 2011) and Stanford Background (StanfordBG) (Gould, Fulton, and Koller 2009) which have 91.6% and 99.5% of their pixels labeled. The SIFT Flow Dataset contains 2688 256×256 color images of 33 semantic categories. The StanfordBG dataset is composed of 715 color images with multiple sizes labeled of 8 semantic categories.

We evaluate all compared methods by Pixel Accuracy (*P.A.*) and Class Accuracy (*C.A.*). *P.A.* is the ratio of correct pixels of all images and *C.A.* means the ratio of correct pixels of all categories. All the experiments are conducted in the transductive learning scheme (Vapnik and Vapnik 1998) which means that the images with sparse annotations are target images. λ is defined as the ratio of picked annotations to total ones. All the applied labels are randomly selected and the quantity of them is equal in every image. Patch labeling is an engineering trick which means a labeled pixel would be covered by a encircling patch indicating the same category although annotators only label the centering pixel. Except for the centering pixel, the surrounding pixels do not contribute to λ as the annotation time of patch labeling and pixel labeling is the same in this way. β is defined as the size of patch where $\beta = 5$ represents one labeled pixel would be covered by a 5×5 patch and the default setting of β is 1.

Algorithm 1 The C^{3}NN algorithm

```
Input:
$I$ = collection of input images;
$L$ = sparse matrix of annotated labels;
$\{K^t\}$= sequence of number of raw clusters where $\{K^t\} = \{K^1, K^2, ..., K^*\}$;
Output:
$Y^*$= final outputs of the framework;
$\Theta^*$= final parameters of CNN;
apply SLIC to divide image set $I$ into superpixel set $Z$;
initialize $X^1$ by making matrix-sized $Z$ into vecter-size data points;
initialize $Y^1$ by clustering $X^1$ into $K^1$ raw clusters and merging them into scene clusters;
initialize parameters of CNN $\Theta^1$ ;
$t \leftarrow 1$
while $K^t < K^*$ do
    update $\Theta^t$ to $\Theta^{t+1}$ by training CNN with labels $Y^t$;
    update $X^t$ to $X^{t+1}$ by feeding $Z$ into CNN;
    get clustering set $C^t$ by clustering $X^{t+1}$ into $K^t$ clusters;
    update $Y^t$ to $Y^{t+1}$ by introducing $L$ to revise $C^t$;
    $t \leftarrow t + 1$;
return $Y^* \leftarrow Y^t$;$\Theta^* \leftarrow \Theta^t$;
```

Experiment setting

In this paper, all the images have been segmented into superpixels with the size of 7×7 and patches extracted from each superpixel are in the size of 27×27 and 45×45. Then, downsample the patches with 45×45 pixels into 27×27 pixels by Gaussian pyramid. We have designed a novel CNN fed with 27×27 image patches by extending the typical LeNet-5 (LeCun et al. 1998) where there are 7 layers. Except for the sensitive analysis of extraction layer, we define the final layer, i.e. layer F7, as the feature extraction layer.

In the experiments, the framework begins with clustering data points of each subspace into 60 raw clusters and the number of clusters would decrease exponentially until meeting the desired number, i.e. final K^* assigned as 15. Other than convergence analysis, final results would be achieved in three rounds. C^{3}NN is fed with 40 images per time. As the model begins with the random selection of labeled pixels, all the experiments were repeated several times to get the average performance. The network is implemented in MATLAB on MatConvNet using a 4-core Intel i7 and GTX 1050 for acceleration. sGAN refers to the semi-supervised GAN

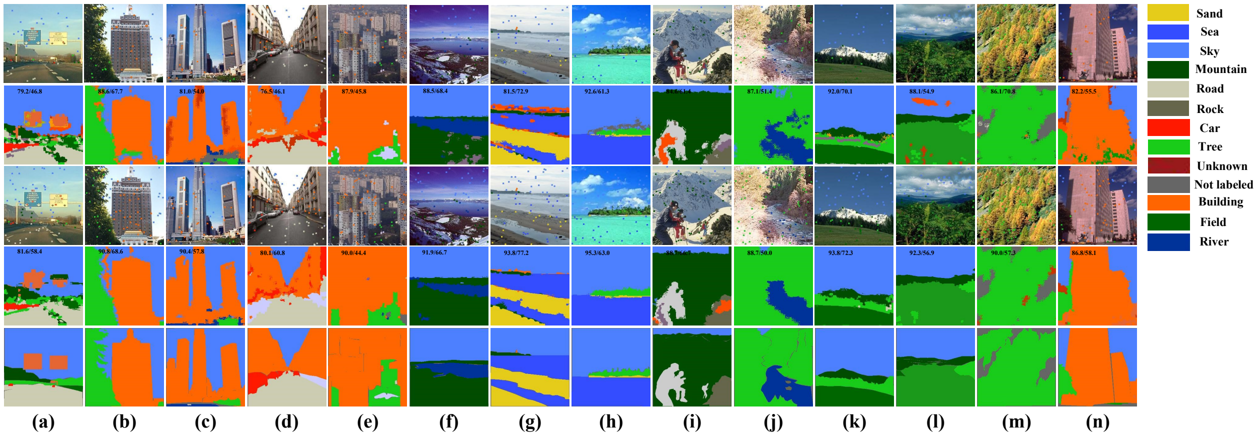


Figure 4: Examples of scene labeling results of C^{3}NN on SIFT Flow dataset. From (a) to (n): fourteen example images. From the top row to the last row: raw input images with sparse labels (λ=0.0005); outputs of C^{3}NN (λ=0.0005,β=5); raw input images with sparse labels (λ=0.001); outputs of C^{3}NN (λ=0.001,β=5); ground truth. The colorful spots represent the applied pixels in the experiments and the images below are the correspondent labeling results of C^{3}NN given these sparse annotations. The category 'not labeled' indicates the regions unlabeled by annotators and 'unknown' represents the regions unrecognized by machines. The printed number on output images are *P.A./C.A.*(%) per image.

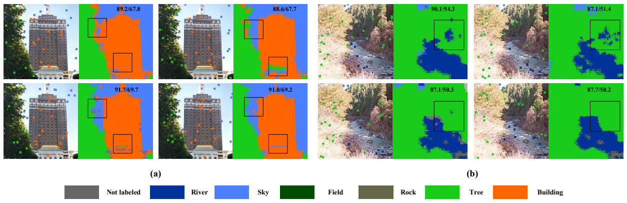


Figure 5: Performance of C^{3}NN (λ=0.0005, β=5) on two example images given four kinds of annotations. The printed number on output images are *P.A./C.A.*(%) per image.

cited from (Souly, Spampinato, and Shah 2017). CSKmeans refers to the Constrained Seed k-means proposed by (Basu, Banerjee, and Mooney 2002).

Quantitative analysis and comparison

Table 1 summarizes the quantitative comparisons of SL performance on the SIFT Flow and StanfordBG dataset. On SIFT Flow dataset, it is noticeable that C^{3}NN(λ=0.005, β=5) outperforms the state-of-the-art method carried in fully supervised manners by 0.9% and 1.9% in *P.A.* and *C.A.* respectively and it is better than semi-supervised GAN which nevertheless exploits more annotations. Moreover, C^{3}NN outperforms CSKmeans in all ways given the same quantity of applied annotation. By applying the patch labeling, we successfully boost the *P.A.* from 70.7% to 81.2% and *C.A.* from 45.9% to 49.8% when λ is 0.0005. The robustness of C^{3}NN has been demonstrated by the minimum deviation through repeated experiments.

On StanfordBG, C^{3}NN(λ=0.005, β=5) is inferior to the state-of-the-art method carried in fully supervised manners by 6.0% in *P.A.* and 0.8% in *C.A.* sGAN outperforms C^{3}NN (λ=0.005, β=5) slightly in *P.A.* while exploiting far more annotations, i.e. 7%. CSKmeans is largely inferior to C^{3}NN and unable to be put into practice because of poor results.

The computation time of C^{3}NN for labeling a 256×256 image is 49.8 seconds on a CPU and 37.2 seconds with a G-PU for acceleration. Multi-CNN and Multi-rCNN take 60.5 seconds and 10.7 seconds per image. The inference time of FCN and FCN+H+LC are 3 seconds and 9 seconds. While considering annotation time, C^{3}NN is the most efficient one as it takes only dozens of seconds for annotating an image.

Figure 4 demonstrates the visual comparison between C^{3}NNs with different values of λ. C^{3}NN (λ=0.0005, β=5) is capable of labeling more than 80% pixels correctly given 33 labeled pixels on a 256×256 image and C^{3}NN (λ=0.001, β=5) could boost it to almost 90% given 65 marked pixels. The annotation burden has been greatly relieved through the proposed method which is able to meet the demands of labeling VIoT data. However, some of tiny objects, such as the building of image (g), little person of image (i) and rocks of image (j) etc., are challenging for C^{3}NN to discriminate them as they are labeled with minimum amounts of annotations. We suggest annotators pay more attentions to tiny objects and mark more labels on them.

We select the image (b) and (j) of Figure 4 for further analysis. As shown by Figure 5, although initialed with different sparse annotations, C^{3}NN performs well and stably. However, the positions of annotations do influence the performance. As indicated by the boxes, some parts of the objects would be mistakenly recognized because of the similarities, especially the colors, between different objects. We strongly recommend the annotators mark these regions in practice.

Sensitivity analysis

There are five types of parameters, i.e. λ, number of recurrent steps, β, K^* and feature extraction layer, required to be examined further to understand mechanism of C^{3}NN.

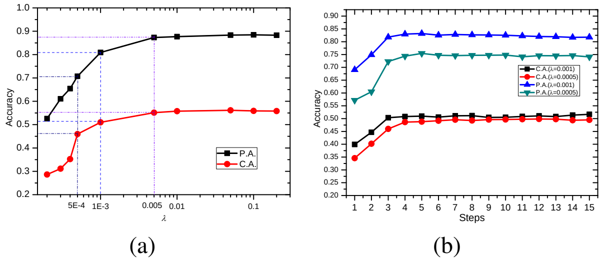
Table 1: Quantitative results and comparison on SIFT Flow and StanfordBG dataset.

Methods	SIFT Flow $P.A._{std}/C.A._{std}(\%)$	StanfordBG $P.A._{std}/ C.A._{std}(\%)$
SuperParsing*	76.3/28.8	-/-
Multi-CNN*	78.5/29.6	81.4/76.0
Multi-rCNN*	77.7/29.8	80.2/69.9
FCN*	85.9/53.9	-/-
FCN+H+LC*	87.8/52.0	**88.2/84.3**
sGAN(R=0.07)	-/-	82.3/77.6
sGAN(λ=0.5)	81.0/33.0	-/-
CSKmeans(λ=0.0005)	$49.8_{1.0}/39.7_{1.2}$	$52.5_{1.8}/52.9_{1.3}$
CSKmeans(λ=0.001)	$52.1_{0.7}/42.6_{1.1}$	$54.0_{0.5}/55.7_{0.9}$
CSKmeans(λ=0.005)	$62.3_{0.3}/52.1_{1.0}$	$63.3_{0.2}/65.2_{0.3}$
C^3NN(λ=0.0005)	$70.7_{0.6}/45.9_{2.1}$	$66.2_{0.5}/69.2_{0.7}$
C^3NN(λ=0.001)	$81.0_{0.4}/50.9_{1.3}$	$74.1_{0.4}/75.8_{0.6}$
C^3NN(λ=0.005)	$88.0_{0.1}/55.1_{1.0}$	$80.9_{0.2}/82.2_{0.3}$
C^3NN(λ=0.0005,β=5)	$81.2_{0.3}/49.8_{2.0}$	$73.6_{0.3}/76.0_{0.4}$
C^3NN(λ=0.001,β=5)	$86.0_{0.1}/54.6_{1.7}$	$78.9_{0.2}/80.1_{0.3}$
C^3NN(λ=0.005,β=5)	$\mathbf{88.7}_{0.1}/\mathbf{55.8}_{0.9}$	$82.2_{0.1}/83.5_{0.3}$

* represents fully-supervised methods. SuperParsing (Tighe and Lazebnik 2010) refers to MRF with superpixels. Multi-CNN (Farabet et al. 2013) is trained with multi-scale CNN. Multi-rCNN (Pinheiro and Collobert 2014) presents the multi-scale, recurrent CNN. FCN (Long, Shelhamer, and Darrell 2017) means the Fully Convolutional Networks. FCN+H+LC (Wang et al. 2017) is trained by Deep CNN with hierarchical loss and label map clustering. C^3NN (λ=0.0005) is feed with 0.05% labeled pixels and C^3NN (λ=0.0005, β=5) is feed with 0.05% 5×5 labeled patches. All the experiments have been repeated ten times with deviations listed as subscripts.

Parameter λ is defined as the ratio of annotated pixels of all pixels per image. The quantitative analysis of parameter λ helps us explore the performance of C^3NN in utilizing sparse annotated labels. As shown in Figure 6 (a), the value of parameter λ would significantly influence the performance. We select three typical values of λ which are 0.0005, 0.001 and 0.005. Before 0.005, the *P.A.* and *C.A.* are directly proportional to the value of λ. Then, it tends to keep constant even increasing the value of λ. 0.005 is a recommended value while it still need too many annotations. Assigning λ as 0.001 or 0.0005 is an alternative choice.

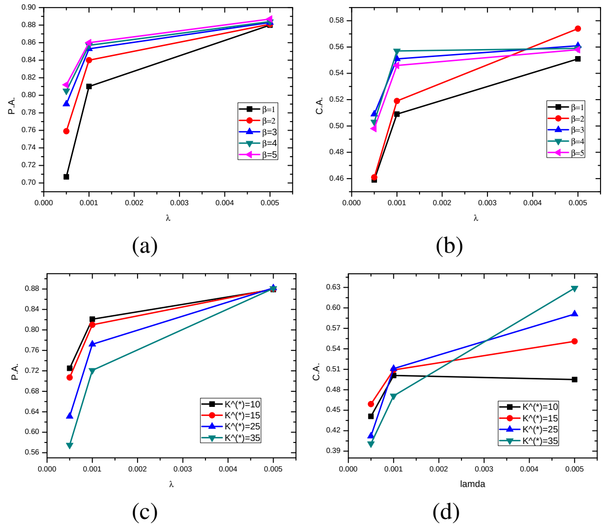
The convergence analysis on C^3NN is necessary for exploring the termination condition of C^3NN. In the Figure 6 (b), both *P.A.* and *C.A.* in the two settings would sharply



(a)

(b)

Figure 6: (a) Performance of C^3NN (β=1) on SIFT Flow dataset in exploiting different ratios of annotations. (b) Convergence analysis of C^3NN (β=1) on SIFT FLow dataset.



(a)

(b)

(c)

(d)

Figure 7: Performance of C^3NN on SIFT Flow dataset in different values of β or K^*. (a) and (b) mean *P.A.* and *C.A.* of C^3NN in different values of β while increasing λ. (c) and (d) mean *P.A.* and *C.A.* of C^3NN (β=1) in different values of K^* while increasing λ.

rise in first three steps and keep stable with little fluctuation then. In first three steps, the accuracy has been largely influenced by the process of cluster merging, while in later steps, raw clusters tend to expand, and therefore retain errors uncorrected through merging. In all, three steps are enough to clearly distinguishing scene clusters.

Patch labeling is an important engineering trick. As shown by Figure 7 (a) and (b), patch size would influence the performance in a large scale. It seems that the largest patch is the best choice. However, the patches marked at the edges or near them would blur the edges in output maps as a large patch could cover more than one object while indicating only one category. There are trade-offs between the *P.A.* and *C.A.*. In this way, β=5 or β=4 both are recommended.

The desired number of clusters, i.e. K^*, is another influential parameter. As shown in Figure 7 (c) and (d), it seems that a larger K^* is helpful for improving *C.A.* and a smaller K^* does good for boosting *P.A.*. While an extremely large or small K^* is useless. K^* is suggested to be one or two times larger than the quantity of semantic categories in an image. While, given different quantity of annotations, the best value of K^* varies. In most of cases, λ=0.0005 and λ=0.001 are acceptable as requiring labeling only tens of pixels. The average number of existed categories per image on SIFT Flow is 5.43. Assigning K^* between 10 and 15 is recommended.

In Table 2, C5, F6, F7 and SoftmaxL indicate the 5-th convolution layer, the 6-th fully connected layer, the 7-th fully connected layer and the Softmax layer where the sizes of features are 1000×1, 200×1, 34×1 and 34×1. The performance of F6 and F7 outperform those of C5 and SoftmaxL in most of cases. It might due to the fact there are more redundant information other than semantic attributes existed at the C5 which is unfriendly for clustering. Meanwhile, there

Table 2: Sensitive analysis of feature extraction layer of C^{3}NN (β=1) on SIFT Flow dataset

λ	C5	F6	F7	SoftmaxL
0.0005	70.2/44.9	**71.4**/45.5	70.7/**45.9**	69.8/42.2
0.001	80.7/49.8	**82.9**/50.2	81.0/**50.9**	78.7/46.0
0.005	87.9/**56.3**	**88.2**/54.9	88.0/55.1	86.6/50.9

The data in each cell indicates *P.A./C.A.%*.

are far less semantic attributes at the Softmax layer as most of them have been weakened in order to outstand the output label. F6 and F7 are best choices for feature extraction.

Conclusion

In this paper, we propose a semi-supervised SL framework, i.e. Constrained Clustering Based Convolutional Neural Network (C^{3}NN), to alleviate the heavy burden of manually annotating. The strength of C^{3}NN lies in its efficiency and accuracy. C^{3}NN is able to correctly label more than 80%, even 90%, pixels given tens of annotations per image in dozens of seconds, while some of tiny objects are easily to be ignored. In most of cases, 90% correct SL results are able to meet the demands of VIoT data analysis like object or abnormal action detection. If researchers need 100% accurate annotations, we suggest correct the outputs of C^{3}NN manually which would take far less time than densely annotating as well. C^{3}NN is different to interactive image segmentation as C^{3}NN aims at labeling the full image while most of interactive methods are designed to label the regions of interest.

The theoretical contribution is that the proposed method introduces a novel scheme by combining representation learning and scene clustering. The process of recurrent optimizations is helpful for understanding the mechanism of DL. Moreover, sparse annotation is an useful and practical type of weak constraints which is worth furthering exploration.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 61772393, 61422209 and 61702392, in part by the National Program for Support of Top-notch Young Professionals of China, in part by the National Key Research and Development Program of China under Grant 2017YFB0802200, and in part by the Fundamental Research Funds for the Central Universities under Grant JBX170311. In addition, we are grateful to the valuable comments and supports of Dr. Puzhao Zhang, Dr. Zedong Tang and Dr. Shanfeng Wang.

References

Achanta, R.; Shaji, A.; Smith, K.; Lucchi, A.; Fua, P.; and Süsstrunk, S. 2012. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34(11):2274–2282.

Basu, S.; Banerjee, A.; and Mooney, R. 2002. Semi-supervised clustering by seeding. In *ICML*, 27–34.

Bearman, A.; Russakovsky, O.; Ferrari, V.; and Fei-Fei, L. 2016. Whats the point: Semantic segmentation with point supervision. In *ECCV*, 549–565.

Bengio, Y.; Courville, A.; and Vincent, P. 2013. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35(8):1798–1828.

Caesar, Holger, U. J., and Ferrari, V. 2016. Region-based semantic segmentation with end-to-end training. In *ECCV*, 381–397.

Chen, L. C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; and Yuille, A. L. 2014. Semantic image segmentation with deep convolutional nets and fully connected crfs. *Computer Science* (4):357–361.

Farabet, C.; Couprie, C.; Najman, L.; and LeCun, Y. 2013. Learning hierarchical features for scene labeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35(8):1915–1929.

Gould, S.; Fulton, R.; and Koller, D. 2009. Decomposing a scene into geometric and semantically consistent regions. In *ICCV*, 1–8.

Gubbi, J.; Buyya, R.; Marusic, S.; and Palaniswami, M. 2013. Internet of things (iot): A vision, architectural elements, and future directions. *Future generation computer systems* 29(7):1645–1660.

Hsu, C.-C., and Lin, C.-W. 2017. Cnn-based joint clustering and representation learning with feature drift compensation for large-scale image data. *arXiv preprint arXiv:1705.07091*.

Kwak, S.; Hong, S.; and Han, B. 2017. Weakly supervised semantic segmentation using superpixel pooling network. In *AAAI*, 4111–4117.

LeCun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86(11):2278–2324.

Liu, C.; Yuen, J.; and Torralba, A. 2011. Nonparametric scene parsing via label transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33(12):2368–2382.

Lloyd, S. 1982. Least squares quantization in pcm. *IEEE Transactions on Information Theory* 28(2):129–137.

Long, J.; Shelhamer, E.; and Darrell, T. 2017. Fully convolutional networks for semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39(4):640–651.

Lowe, D. G. 2004. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision* 60(2):91–110.

Ma, H.; Liu, L.; Zhou, A.; and Zhao, D. 2016. On networking of internet of things: Explorations and challenges. *IEEE Internet of Things Journal* 3(4):441–452.

Pan, T.; Wang, B.; Ding, G.; and Yong, J.-H. 2017. Fully convolutional neural networks with full-scale-features for semantic segmentation. In *AAAI*, 4240–4246.

Papandreou, G.; Chen, L.-C.; Murphy, K. P.; and Yuille, A. L. 2015. Weakly-and semi-supervised learning of a deep convolutional network for semantic image segmentation. In *ICCV*, 1742–1750.

Pinheiro, P., and Collobert, R. 2014. Recurrent convolutional neural networks for scene labeling. In *ICML*, 82–90.

Rumelhart, D. E., and Mcclelland, J. L. 1986. *Parallel Distributed Processing*. The MIT Press,.

Souly, N.; Spampinato, C.; and Shah, M. 2017. Semi and weakly supervised semantic segmentation using generative adversarial network. *arXiv preprint arXiv:1703.09695*.

Tighe, J., and Lazebnik, S. 2010. Superparsing: Scalable nonparametric image parsing with superpixels. In *ECCV*, 352–365.

Triggs, B., and Verbeek, J. J. 2008. Scene segmentation with crfs learned from partially labeled images. In *NIPS*, 1553–1560.

Vapnik, V. N., and Vapnik, V. 1998. *Statistical learning theory*. Wiley New York.

Wagstaff, K.; Cardie, C.; Rogers, S.; and Schrödl, S. 2001. Constrained k-means clustering with background knowledge. In *ICML*, 577–584.

Wang, Z.; Li, H.; Ouyang, W.; and Wang, X. 2017. Learning deep representations for scene labeling with guided supervision. *arXiv preprint arXiv:1706.02493*.

Yang, J.; Parikh, D.; and Batra, D. 2016. Joint unsupervised learning of deep representations and image clusters. In *CVPR*, 5147–5156.

Zhang, Z., and Huang, M. 2017. Discriminative structural metric learning for person re-identification in visual internet of things. *IEEE Internet of Things Journal*.

Zhang, X.; Wang, X.; and Jia, Y. 2013. The visual internet of things system based on depth camera. In *Chinese Intelligent Automation Conference*, 447–455. Springer.