Appl Intell (2014) 40:389–403
DOI 10.1007/s10489-013-0473-1

Image annotation by modeling Supporting Region Graph

Qiao-Jin Guo · Ning Li · Yu-Bin Yang · Gang-Shan Wu

Published online: 13 September 2013

Abstract Annotating image regions with keywords has received increasing attention in the computer vision community in recent years. Recent studies have shown that graphical modeling techniques, such as Conditional Random Fields (CRF), greatly improves the accuracy of image annotation by utilizing contextual information among image regions. However, training and predicting with the high-order CRF is computational expensive so that only adjacent regions can be utilized to build its graph structure. In this paper, we develop a light-weight classification model, Approximated Supporting Region Graph (ASRG), in order to handle more relevant regions efficiently, with which a large number of supporting regions are selected and their features are utilized to represent the contextual information in the training and prediction for each image region. Experimental results show that our model is much more computational efficient and achieves competitive performance comparing with CRF and other state-of-art methods.

Keywords Image annotation · Context · CRF · Image segmentation

Q.-J. Guo · N. Li (✉) · Y.-B. Yang · G.-S. Wu
National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China
e-mail: ln@nju.edu.cn

Q.-J. Guo
e-mail: guoqiaojin@gmail.com

Y.-B. Yang
e-mail: yangyubin@nju.edu.cn

G.-S. Wu
e-mail: gswu@nju.edu.cn

1 Introduction

Image annotation techniques assign metadata, usually keywords, to images automatically, which makes it easier for indexing and maintaining large collections of images and thus has been studied actively in the last few decades, particularly in image retrieval [40]. Region-based image annotation, also known as region-naming, region-labeling, and multi-class image segmentation, is one of the most important methods for image annotation. Consequently, various machine learning techniques have been employed for learning the correspondence between image regions and textual keywords [5, 14, 16, 29, 35].

For region-based image annotation, each image is annotated with a set of keywords associated with their locations in the image. Figure 1 shows a sample image from the 21-class MSRC dataset [7], in which each pixel is associated with one of the 21 classes, or a "void" class. There are usually two steps in region-based image annotation: (1) images are segmented into several regions and visual features are extracted from each region; and (2) statistical models are estimated with training samples and then each region is classified into different classes.

However, visual features of each region are not sufficiently discriminative for classification. Contextual information, such as spatial interactions, has shown great success

Fig. 1 A sample image in MSRC (the *first column*) with the ground truth annotation (the *second column*) and over-segmented superpixels (the *right column*)

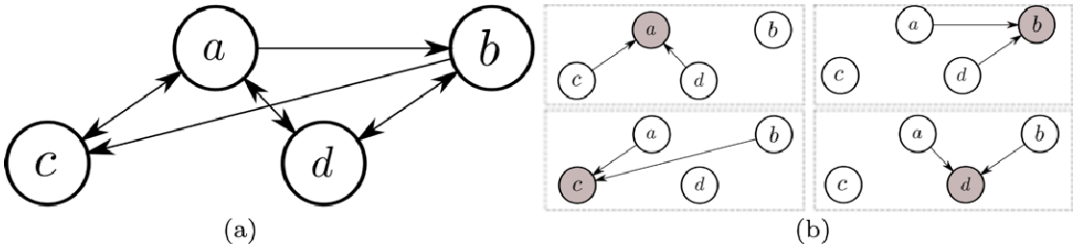


Fig. 2 A simple example of supporting region graph (SRG) with four regions

in image annotation. One of the most frequently used techniques for capturing spatial interactions among image regions is Conditional Random Fields (CRF) [12, 14], which constructs an undirected graph structure by linking adjacent regions. High order interactions are usually ignored due to its heavy computational burden.

In this paper, we propose a new graph model, Supporting Region Graph (SRG), to handle more possible interactions. Different from CRF, we propose a directed graph model, in which edges are added by selecting a set of regions called as "supporting regions" for each region in the image. Figure 2(a) shows a simple example of SRG with four regions $\{a, b, c, d\}$. As shown in Fig. 2(b), each region has two different supporting regions, and the directed edge shows the relationship between each center region and its supporting regions, e.g., the center region 'a' has two supporting regions 'c' and 'd'. The SRG is finally constructed if all directed edges have been added into the same graph. The details of the construction of SRG will be discussed in Sect. 4.

The drawback of CRF is that training with large or high order graph structures is computationally expensive. With the increase of the number of supporting regions, the SRG will encounter the same problem. Therefore, we further propose an approximated version of SRG (ASRG) and apply it on three natural datasets to carry out our experiments. The experimental results in Sect. 5 show that our approach is more efficient and achieves better or similar performance than CRF and other state-of-art methods.

2 Related work

Image Annotation is one of the most important problems in computer vision and has received increasing attention in the last few years. One of the most frequently used techniques is building statistical models on local appearance features [3, 6, 22, 24, 28]. However, the spatial interactions and other relationships among different regions are ignored when training and predicting with each region independently. Contextual features, such as co-occurrence and spatial interactions, have then been utilized to improve classification performance in computer vision applications [25, 26, 32, 38]. Various machine learning techniques have been employed to model contextual relationships between different regions [8, 17, 19–21, 31, 34].

Markov Random Fields (MRF) and Conditional Random Fields (CRF) are two common approaches to capture the spatial relationships among neighborhood regions. Rabinovich et al. [26] used local detectors, assigning an object label to each segmented region, and then adjusted the labels using CRF. Semantic object contexts could then be incorporated as a post-processing step in any off-the-shelf object categorization model. Verbeek et al. [36] extended the topic model with MRF over the latent topics to combine the statistical co-occurrences of quantized visual features and spatial relationships. He et al. [14] utilized multi-scale Conditional Random Fields to annotate image regions by incorporating both local and global image features. Gould et al. [12] proposed a method to capture global information from the inter-class spatial relationships and then annotated image regions by combining locally related location features and visual features. Ladicky et al. [18] presented a hierarchical CRF framework to integrate features and contextual priors defined over multiple image segmentations. Nevertheless, both MRF and CRF share the same drawback that training with large or high order graph structures is computationally expensive. Moreover, the complexity also increases as the number of classes grows. As a result, most of the existing studies only considered the immediately adjacent regions when utilizing MRF or CRF, that is, each node in the graph is linked only to very limited number of adjacent nodes. The graph model proposed in this paper intends to accommodate more relevant regions in the training and prediction while avoiding the problem of computational complexity.

The proposed SRG model is based on direct graph, similar to Dependency Network [15], where the directed edges are utilized to capture the conditional distributions among different nodes. However, Dependency Network is employed to model dependencies among different attributes and the graph structure is fixed. Our proposed SRG models the relationships of segmented image regions, thus the graph structure varies on different segmentation results. Tu [33] proposed an iterative model using contextual information obtained from neighboring regions. Each pixel has "support" from a large number of neighbors, either in short or

long range. The learning algorithm selected and fused important supporting contextual pixels automatically and iteratively. Similar to auto-context [33], the graph structure of our proposed model is built by selecting the surrounding regions. However, our graph is constructed based on directed graph model and the supporting contextual regions are selected directly without any iteration, which makes it simple and efficient. Fulkerson et al. [9] presented a method for localizing objects and segmenting object classes. Regions in neighborhood were merged and their histograms were then beaggregated as the appearance features of the merged region used to capture contextual information. This work can be regarded as selecting the supporting regions with a threshold of their spatial distances. Our work generalizes the supporting region selection strategies by selecting the supporting regions using spatial, visual or other properties, such as the predicted class probabilities. Meanwhile, we propose an extended version of ASRG using multiple groups of supporting regions (mASRG) to represent different contextual information.

The contributions of our work can be summarized as follows:

(1) We propose a directed graph model by incorporating contextual information obtained from the selected supporting regions.
(2) We process a fast image annotation framework by designing an approximation of the directed graph model mentioned in (1).
(3) We analyze the annotation accuracy and efficiency of some different strategies for selecting one or multiple groups of supporting regions.

3 Modeling Supporting Region Graph

3.1 Supporting Region Graph

We first formulate the definition of Supporting Region Graph (SRG). Let $\boldsymbol{x} = \{x_1, \ldots, x_i, \ldots, x_N\}$ be the observed data and $\boldsymbol{y} = \{y_1, \ldots, y_i, \ldots, y_N\}$ be the class labels, where x_i, y_i be the observed feature and label of region i and N is the number of nodes in the current graph.

Definition of SRG Let $A = (V, E)$ be a graph and $\boldsymbol{y}$ is indexed by the vertices of A. Then $(\boldsymbol{y}, \boldsymbol{x})$ is defined as a Supporting Region Graph if, when conditioned on $\boldsymbol{x}$, the random variables y_i obey the Markov property with respect to the graph: $p(y_i \mid \boldsymbol{x}, \boldsymbol{y}_{V-\{i\}}) = p(y_i \mid \boldsymbol{x}, \boldsymbol{y}_{S_i})$, where $V - \{i\}$ is the set of all other regions in A except for the region i, and S_i is the set of supporting regions of region i in A.

Similar to CRF, the joint distribution over the class labels $\boldsymbol{y}$, given the observations $\boldsymbol{x}$, can be expressed as:

$$p(\boldsymbol{y} \mid \boldsymbol{x}) = \frac{1}{Z} \exp\left(\sum_{i=1}^{N} F(x_i, y_i) + \sum_{i=1}^{N} \sum_{j \in S_i} \frac{1}{|S_i|} G(x_i, x_j, y_i, y_j) \right) \tag{1}$$

$$Z = \exp\left(\sum_{i=1}^{N} \sum_{y'_j=1}^{M} F\left(x_i, y'_i\right) + \sum_{i=1}^{N} \sum_{j \in S_i} \sum_{y'_i=1}^{M} \sum_{y'_j=1}^{M} \frac{1}{|S_i|} G\left(x_i, x_j, y'_i, y'_j\right) \right) \tag{2}$$

where $|S_i|$ is the number of elements of S_i, M is the number of classes, Z is the partitioning function and y_i, y'_i are the labels of node i. F is the node potential and G is the edge potential.

$$F(x_i, y_i) \propto P(y_i \mid x_i) \tag{3}$$

$$G(x_i, x_j, y_i, y_j) \propto P(y_i \mid x_i, x_j, y_j) \tag{4}$$

where $P(y_i \mid x_i)$ is the probability of having label y_i with observed feature x_i, $P(y_i \mid x_i, x_j, y_j)$ is the edge probability, x_i, y_i is the feature and label of node i, and x_j, y_j is the feature and label of node j, where $j \in S_i$.

As shown in Fig. 2, the conditional probability of node 'a' depends on the node potential of itself and the edge potentials of its supporting regions 'c' and 'd'. Thus, $p(y_a \mid x_a, x_c, x_d) \propto F(x_a, y_a) + (G(x_a, x_c, y_a, y_c) + G(x_a, x_d, y_a, y_d))/2$. The edge potential G captures the interactions between region 'a' and its supporting regions $\{c, d\}$, which are ignored in traditional non-structured classifiers, such as SVM and Logistic Regression. The classification performance of node 'a' can be improved by modeling relationships between 'a' and $\{c, d\}$. Structured classifiers, such as CRF, also employ node potentials to capture spatial interactions between different regions. Different from CRF, SRG is a directed graph model, thus $G(x_a, x_c, y_a, y_c) \neq G(x_c, x_a, y_c, y_a)$.

The training process of SRG contains two parts, (1) estimating node probabilities and (2) estimating edge probabilities. Given D training samples, each sample contains N_d, $d = 1, \ldots, D$ regions and each region has k supporting regions. To estimate node probability $P(y_i \mid x_i)$, we need to train a M-class classifier with $\sum_{d=1}^{D} N_d$ regions, in which M is the number of different classes. To calculate edge probability $P(y_i \mid x_i, x_j, y_j)$, we need to train the edge classifiers with $\sum_{d=1}^{D} kN_d$ edge instances, which is time consuming

and the complexity increases rapidly as k grows. Considering SRG is a directed graph, we split all directed edges into M groups with different y_j. Afterwards, we train a M-class edge classifier for each group, using x_i, x_j as features and y_i as the class label. Thus, for each edge in the tested samples, we generate a probability matrix P_e with the size of $M \times M$, in which $P_e(g,h) = P(y_i = g \mid x_i, x_j, y_j = h)$. After all nodes and edge classifiers are obtained, we employ loopy belief propagation (LBP) to perform inference over SRG on test samples.

The number of training instances increases rapidly with the number of supporting regions k. For each edge group, the edge classifier has approximately $\sum_{d=1}^{D} kN_d/M$ instances. As discussed above, the more supporting regions, the more contextual information may be captured. However, training SRG with a large k is computational expensive, so is the inference on test samples. To overcome this shortcoming, we further propose an approximated version of SRG (ASRG) to make it much more computational efficient.

3.2 Approximated Supporting Region Graph

Since the number of edges in SRG increases with parameter k, we need to find a more efficient way to train SRG if we need to accommodate more supporting regions. We start from the Maximum Likelihood (ML) of training samples. Let θ be the set of parameters of SRG. Given a training set $\{(\boldsymbol{x}_d, \boldsymbol{y}_d) \mid d = 1, \ldots, D\}$, the likelihood is:

$$\mathcal{L}(\theta) = \prod_{d=1}^{D} p(\boldsymbol{y}_d \mid \boldsymbol{x}_d) \tag{5}$$

Directly maximizing $\mathcal{L}(\theta)$ is intractable as discussed above. Therefore, in order to make it solvable, we first separate SRG into a group of subgraphs. An example partition of SRG is shown in Fig. 2(b), each subgraph consisting one center region and its supporting regions. For each subgraph, we maximize the likelihood:

$$\mathcal{L}_i = \frac{1}{Z} \exp\left(F(x_i, y_i) + \sum_{j \in S_i} \frac{1}{|S_i|} G(x_i, x_j, y_i, y_j) \right) \tag{6}$$

Here we define $F(x_i, y_i) = \boldsymbol{u}_{y_i}^T x_i$, $G(x_i, x_j, y_i, y_j) = \boldsymbol{v}_{y_i y_j}^T x_j$. $\boldsymbol{u}, \boldsymbol{v}$ as the weight vectors of nodes and edges, and $\theta = \{\boldsymbol{u}, \boldsymbol{v}\}$.

However, the labels of supporting regions are unknown in the optimization process. In order to optimize the parameters without knowing the labels of supporting regions, we treat the supporting regions as hidden variables. All configurations of the labels are enumerated, and the probabilities of them are considered as uniform. Then, we can maximize:

$$\hat{\mathcal{L}}_i = \frac{1}{Z} \exp\left(F(x_i, y_i) + \sum_{j \in S_i} \frac{1}{|S_i|} \hat{G}(x_j, y_i) \right) \tag{7}$$

where $\hat{G}(x_j, y_i) = \boldsymbol{v}_{y_i} x_j$ and $\boldsymbol{v}_{y_i} = \frac{1}{m} \sum_{y_j=1}^{m} \boldsymbol{v}_{y_i y_j}$.

Maximizing $\hat{\mathcal{L}} = \prod_{d=1}^{D} \hat{\mathcal{L}}_i$ is much easier than directly maximizing $\mathcal{L}$. The posterior probability is

$$P\big(y_i \mid x_i, \{x_j\}, \theta\big) = \frac{\exp(\boldsymbol{u}_{y_i}^T x_i + \sum_{j \in S_i} \frac{1}{|S_i|} \boldsymbol{v}_{y_i} x_j)}{\sum_{y_i'} \exp(\boldsymbol{u}_{y_i'}^T x_i + \sum_{j \in S_i} \frac{1}{|S_i|} \boldsymbol{v}_{y_i'} x_j)} \tag{8}$$

Thus, given D independent training samples, where each sample contains $N_d, d = 1, \ldots, D$ regions, the parameters of ASRG can be optimized as:

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}} \sum_{d=1}^{D} \sum_{i=1}^{N_d} \big\{ \log P(y_i \mid x_i, s_i, \theta) \big\} \tag{9}$$

where $s_i = \sum_{j \in S_i} \frac{1}{|S_i|} x_j$ and we define s_i as the supporting features of region i. The parameters θ can be optimized using a process similar to Logistic Regression, which could be optimized using stochastic gradient descent (SGD) methods. Actually, we can use any other basic classifier to maximize Eq. (9) with the corresponding $P(y_i \mid x_i, s_i, \theta)$, such as SVM and Random Forest, in which each node is represented with its node features x_i and its supporting features s_i.

Figure 3 shows the differences among CRF, Logistic Regression (LR), SRG and ASRG. CRF utilizes edges between the class labels of different nodes to capture conditional dependence, while LR considers that the conditional probabilities of different nodes are independent. For CRF, it captures the contextual relationships by adding undirected edges between adjacent nodes. Different with CRF, SRG employs a set of selected supporting regions to describe contextual interactions, which contains more long-range edges. ASRG uses supporting features extracted from the selected supporting regions to capture contextual information. Unlike the undirected graph used in CRF, both SRG and ASRG utilize directed graph structures which is able to capture asymmetric co-occurrences. In LR, the class probability of each node is directly conditional on visual features; but in ASRG, the class probability is conditional on the region and the supporting features of its selected supporting regions. Similar to LR, the class probabilities of each node in ASRG is independent to each other. This is different with SRG and CRF and thus makes the training and inference processes faster.

3.3 Modeling multiple groups of supporting regions

In the classification for each image region, the relationships between each supporting region and the center region may be different. For example, the center region is with class "face", but some supporting regions may be with class "face", or class"body" and even some other classes. In order

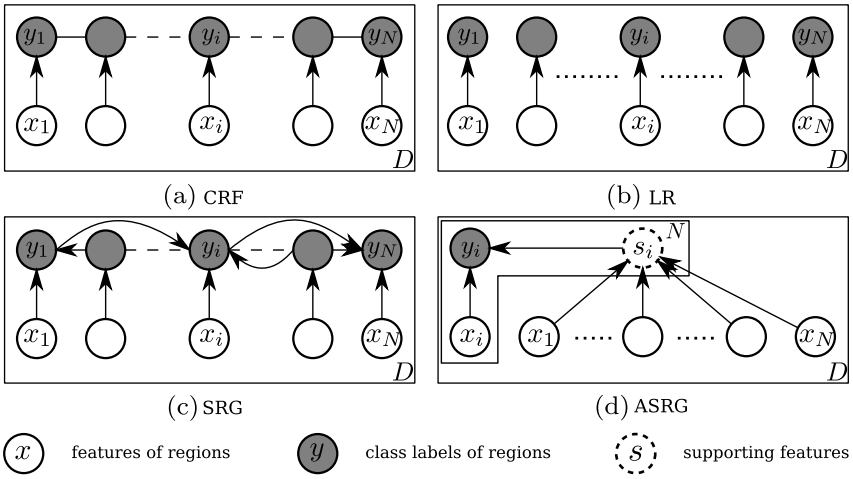


Fig. 3 The differences among CRF, LR, SRG and ASRG. x represents the features of each region, y represents the corresponding class labels, s represents the supporting features extracted from the selected supporting regions, D is the number of training samples, and N is the number of regions in each sample

to capture this characteristic, we further separate the supporting regions into different groups, not necessarily non-intersect, and assign different weight vector for each group. Note that the regions of each group share the same weight vector, and the weight vectors of different groups are independent. For each region i, a number of supporting region groups $\{S_i^k \mid k = 1, \ldots, K\}$ are selected and used to construct a complex directed graph, where v^k is the edge weighting vector of each group. The likelihood of classifying with multiple groups of supporting regions is:

$$P(\boldsymbol{y} \mid \boldsymbol{x}) = \frac{1}{Z} \exp\left(\sum_{i \in N} F(x_i, y_i) + \sum_{i=1}^{N} \sum_{k=1}^{K} \sum_{j \in S_i^k} \frac{G^k(x_j, y_i, y_j)}{|S_i^k|} \right) \tag{10}$$

where F_i is the node potential of node i, G_{ij}^k is the edge potential of edge ij in the kth group, and $G^k(x_j, y_i, y_j) = \boldsymbol{v}_{y_i y_j}^k x_j$. $|S_i^k|$ is the number of elements in S_i^k.

Thus, the likelihood of multi-group Approximated Supporting Region Graph (mASRG) can be computed as:

$$P(y_i \mid x_i, s_i, \theta) = \frac{\exp\left(\begin{bmatrix} \boldsymbol{u}_{y_i} \\ \boldsymbol{v}_{y_i}^1 \\ \vdots \\ \boldsymbol{v}_{y_i}^K \end{bmatrix}^T \begin{bmatrix} x_i \\ s_i^1 \\ \vdots \\ s_i^K \end{bmatrix} \right)}{\sum_{y_i'} \exp\left(\begin{bmatrix} \boldsymbol{u}_{y_i'} \\ \boldsymbol{v}_{y_i'}^1 \\ \vdots \\ \boldsymbol{v}_{y_i'}^K \end{bmatrix}^T \begin{bmatrix} x_i \\ s_i^1 \\ \vdots \\ s_i^K \end{bmatrix} \right)} \tag{11}$$

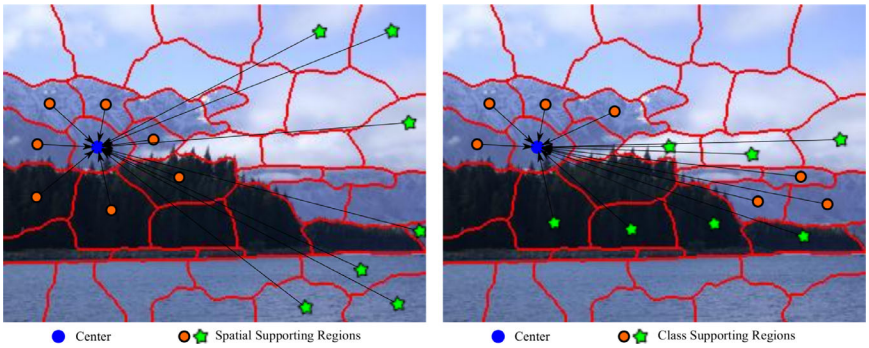
Thus each image region can be represented with its own features x_i and its supporting features $\{s_i^k \mid k = 1, \ldots, K\}$ from different supporting groups.

4 Supporting region selection in image annotation

In this Section, we present how the supporting region graph is constructed for image annotation. For each region, we need to select several relevant regions from its surrounding regions. However, it is not that easy to select the supporting regions appropriately. To solve this problem, we use a simple but efficient scheme that we rank the image regions based on the responses of a pre-defined response function $E(f_i, f_j, w)$. As described in Eq. (12), in order to select the supporting regions of region i, we calculate the response of region j corresponding to the center region i, and the weighted Euclidean distance of features between region i and j, where f_j can be the visual feature, the class probabilities, or the spatial location of region j. w is the weighting parameter defining the importance of each feature element and $w \in [-1, 1]^F$, F is the dimension of f.

$$E(f_i, f_j, w) = \left\| w^T (f_i - f_j) \right\|^2 \tag{12}$$

For each center region i, we calculate the response values of all other regions in the same image. After that, these regions are sorted according to $E(f_i, f_j, w)$. Usually, we can select k regions with the minimum responses as supporting regions. For selecting two groups of supporting regions, we can use the regions with the minimum and the maximum response values. Figure 4 shows two examples of selecting supporting regions with spatial and class responses. As shown in Fig. 4(a), 7 nearest and 7 farthest regions are selected for the center region by utilizing spatial locations. Figure 4(b) shows an example of selecting supporting regions which uses class probabilities as the input features of Eq. (12), in which 7 regions having similar class probabilities with the center region are selected, and 7 other regions with the largest response values are also selected as another group. In this paper, we present 6 different strategies for supporting region selection, including class-kN, class-kNF,



(a) Spatial Supporting Regions (b) Class Supporting Regions

Fig. 4 Examples of supporting region selection with spatial and class response functions

spatial-kN, spatial-kNF, visual-kN, and visual-kNF. The details are defined in Table 1.

In CRF, each region is linked with its spatial neighbors and the number of links is usually small. In our experiments, each region in CRF is linked with 5 to 6 regions averagely. Selecting supporting regions with spatial-kN is similar to CRF, in which useful contextual information can be extracted from the neighborhood relationships and utilized to improve classification performance. However, in ASRG, the number of links can be very large and more interactions among regions can be captured during annotation. By selecting supporting region with spatial-kNF, not only neighboring regions are selected, but also regions in long ranges are used in training and prediction. Consequently, if the nearest supporting regions are viewed as "foreground", the farthest regions can be regarded as "background", by which useful contextual information can be extracted from the relationships between "foreground" and "background" regions and further utilized to improve the classification performance.

Selecting supporting regions with visual features is based on the assumption that regions with similar features tends to share similar labels in the same image, and regions with different visual features usually belong to different classes. Selecting supporting regions with class probabilities is similar to the selection strategy with visual features. The selection is more precise than directly selecting with visual features but a pre-trained classifier is employed to select regions with the same or different class labels. The features of the selected nearest regions in visual/class response space are utilized to describe the supporting information from similar classes, while the selected farthest regions may capture the co-occurrences between different classes.

By changing the weights w, we may obtain different supporting regions. In this paper, we use two different weighting schemes: (1) directly set $w = 1$; and (2) optimize w with the training images. In order to optimize w, the training images are separated into two sets. After that, we use different ws to select supporting regions in one set and test the annotation accuracy on the other. We employ genetic algorithm (GA) to search the best w and use the annotation accuracy as the fitness function. Searching w with GA is time consuming, thus we may also directly set $w = 1$ for efficiency purpose. Experimental results show that selecting supporting regions with $w = 1$ achieves similar performances with the optimized w when k is large.

Table 1 Definitions of different supporting region selection strategies

Distance measure	Details
Class	Output probabilities of each region utilizing a pre-trained classifier
Spatial	Center coordinates of each region
Visual	Visual features of each region
Number of SR	**Details**
kN	selecting k nearest regions in the response space
kNF	selecting $k/2$ nearest and $k/2$ farthest regions in the response space

An extreme situation is to use all regions in an image as supporting regions. We define this special model as Extreme Approximated Supporting Region Graph (EASRG). The performance of those strategies will be evaluated in the experiment section.

5 Experiments

5.1 Datasets

We applied our model on annotating three natural image datasets, including the 7-class Corel image database, the 9-class and the 21-class MSRC dataset [7]. The Corel image dataset consists 100 images, in which each image is 180×120 pixels and includes objects like "*rhino, polar bear, water, snow, vegetation, ground and sky*". We selected 60 images for training and the remaining 40 for testing. The 9-class MSRC dataset contains 240 pixel-wise annotated images with approximately 240×320 pixels, including objects like "*building, grass, tree, cow, sky, aeroplane, face,*

car and bicycle". The 21-class MSRC dataset contains 591 pixel-wise annotated images with approximately 240 × 320 pixels, including objects like "*building, grass, tree, cow, sheep, sky, aeroplane, water, face, car, bicycle, flower, sign, bird, book, chair, road, cat, dog, body and boat*". On the 9-class MSRC dataset, we split the images into 120 training and 120 testing. On the 21-class MSRC dataset, we divided the images into a training set with 296 images and a testing set with 295 images.

5.2 Segmentation and feature extraction

First, each image was over-segmented into small regions, each of which was labeled with the dominant class label for the region. Void regions were ignored for both training and testing. We used the SLIC code provided by Achanta [1, 2] to segment each image into approximately 200 regions. Images from the Corel dataset were first resized into 240 × 320 for consistency. After segmentation, the average number of superpixels in the Corel dataset was 161.39. The 9-class MSRC dataset had 174.09 superpixels in each image, and the 21-class MSRC dataset had 175.74 superpixels in each image.

For each region, we extracted local visual features including color, intensity, texture, geometry and location [13]. Color features were extracted from the color values of each image region. The initial RGB image was converted into different color spaces, including YCrCb and Lab. Moreover, a supersaturated RGB image was generated by increasing the saturation of initial image. Thus, there were 4 different images (RGB, YCrCb, Lab and supersaturated RGB) and each contained 3 channels. For each channel, 4 moments features were extracted, including mean, variance, skewness and kurtosis. Intensity features were extracted from the grayscale images. Each image was first converted to grayscale and 4 moments features were extracted from each region similar with Color features.

Texture features were extracted by utilizing Gabor and Laplacian of Gaussian filters. Gabor filters were defined as:

$$\begin{aligned} &G_{\text{real}}(x,y) \\ &\quad = \exp\bigl(-\bigl(X^2+\gamma^2Y^2\bigr)/\bigl(2\sigma^2\bigr)\bigr)\cos\bigl((2\pi X)/\lambda\bigr) \\ &G_{\text{imaginary}}(x,y) \\ &\quad = \exp\bigl(-\bigl(X^2+\gamma^2Y^2\bigr)/\bigl(2\sigma^2\bigr)\bigr)\cos\bigl((2\pi Y)/\lambda\bigr) \end{aligned} \tag{13}$$

where x, y were the coordinates, $X = x\cos(\theta) + y\sin(\theta)$ and $Y = -x\sin(\theta) + y\cos(\theta)$ with orientation θ (in radians), aspect ratio γ, effective width σ and spatial frequency λ. In the following experiments, θ takes $\{0, \pi/4, \pi/2, 3\pi/4\}$, $\gamma \in \{0.25, 1, 4\}$, $\sigma \in \{2, 4, 8\}$ and $\lambda = 1$.

Laplacian of Gaussian (LoG) filters were defined as:

$$LoG(x,y) = 1/\bigl(\pi\sigma^4\bigr)\bigl(r^2-1\bigr)\exp\bigl(-r^2\bigr) \tag{14}$$

where $r^2 = (x^2+y^2)/(2\sigma^2)$. In the following experiments, $\sigma = \{2^{\frac{i}{2}} \mid i = 1, \ldots, 5\}$.

Each image was converted to grayscale and convolved with a bank of filters and momtents features were extracted from the response images. In the experiments, Gabor filters took 36 configurations, LoG filters took 5 configurations, and this resulted in 41 different response images. For each reponse image, 4 moments features (mean, variance, skewness and kurtosis) were extracted. Thus, 164 texture features were extracted for each region in total.

Geometry features were extracted from image regions with the statistical properties of the shapes, including area, perimeter, perimeter area ratio and three moments features of coordinates $\{E(x^2) - E(x)^2, E(y^2) - E(y)^2, E(xy) - E(x)E(y)\}$, where E was the expectation. Location features were extracted by estimating the center coordinates of each image region, including $\{x_{\text{center}}, y_{\text{center}}, x_{\text{center}}^2 + y_{\text{center}}^2\}$.

A total of 225 visual features were extracted from each region, including 48 Color features, 4 Intensity features, 164 Texture features, 6 Geometry features and 3 Location features. According to [13], we trained boosted classifiers for each class and used the outputs of the boosted classifiers as features instead of the raw appearance features. Feature extraction and training boosted classifiers were performed by utilizing STAIR Vision Library [13] provided by Stephen Gould.

The time cost of segmentation and feature extraction are shown in Table 2. All the following experiments were run on a PC with 2.4 GHz Core i5 CPU and 2 G RAM. Using boosted features improves the classification accuracy and reduces the training time as well. Figure 5 shows the comparisons of training time and classification accuracy with raw features (RFS) and boosted features (BFS) on two MSRC datasets. We evaluated two different features with two state-of-art classifiers, random forests (RF) [4] and conditional random fields (CRF). The experimental results show that the training speed and classification accuracy are both improved by utilizing boosted features. On the other hand, BFS can also be used as initial class probabilities to select supporting regions. As a result, in the following experiments, we used boosted features to evaluate different annotation methods.

Table 2 Time costs of over-segmentation and feature extraction (seconds per image)

Steps (seconds per image)	Corel-7	MSRC-9	MSRC-21
Over-segmentation	0.353	0.324	0.311
Feature extraction	0.505	0.470	0.505
Generate boosted features	0.031	0.039	0.060

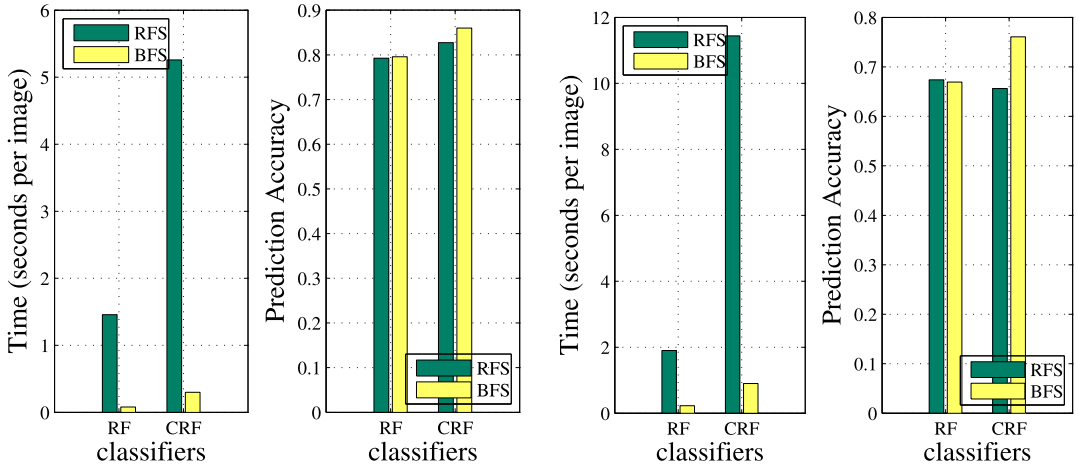


Fig. 5 Comparison of training time and classification accuracy with raw features (RFS) and boosted features (BFS) on the 9-class and 21-class MSRC dataset

5.3 Comparisons of different supporting region selection

Different supporting regions can be selected by utilizing different features. We use c, s, v as the abbreviation of 'class', 'spatial' and 'visual', N is the abbreviation of 'nearest' and F is the abbreviation of 'farthest'.

In order to evaluate the efficiency of the different selecting methods, we compared the annotation accuracy of ASRG with 8 different methods on all three datasets.

(1) c-kN: class-KN, select k nearest regions with class probabilities
(2) c-kNF: class-KNF, select $k/2$ nearest and $k/2$ farthest regions with class probabilities
(3) s-kN: spatial-KN, select k nearest regions with spatial locations
(4) s-kNF: spatial-KNF, select $k/2$ nearest and $k/2$ farthest regions with spatial locations
(5) v-kN: visual-KN, select k nearest regions with visual features
(6) v-kNF: visual-KNF, select $k/2$ nearest and $k/2$ farthest regions with visual features
(7) c-kN-GA: class-KN, select k nearest regions with class probabilities using optimized weights
(8) c-kNF-GA: class-KNF, select $k/2$ nearest and $k/2$ farthest regions with class probabilities using optimized weights

The details of training ASRG are described in Algorithm 1.

We used the boosted features as class probabilities to select class supporting regions, and the raw visual features were utilized to select visual supporting regions. The spatial supporting regions were selected by utilizing the center

Algorithm 1 Processes of training ASRG

Input:
Training images $\{I_d \mid d = 1, \ldots, D\}$
number of supporting regions k
supporting region selecting strategies {N, NF}
response feature types {class, visual, spatial}
Output:
Optimized parameters θ

Apply over-segmentation on training images, where image I_d is segmented into N_d regions
Extract appearance visual features a for each image region
Train boosted features b for each region according to [13]
for Each image $I_d, d = 1, \ldots, D$ **do**
  for Each region $r_{di}, i = 1, \ldots, N_d$ in image I_d **do**
    Extract response features f_{di}
    if using visual features **then**
      $f_{di} = a_{di}$
    else if using class features **then**
      $f_{di} = b_{di}$
    else if using spatial features **then**
      $f_{di} = [x_{di}^{\text{center}}, y_{di}^{\text{center}}]^T$, where $(x_{di}^{\text{center}}, y_{di}^{\text{center}})$ is the center coordinates of r_{id}
    end if
  end for
  Calculate and sort the response values according to Eq. (12) with f_{di}
  if using kN **then**
    Select k supporting regions with smallest response values
  else if using kNF **then**
    Select $k/2$ supporting regions with smallest response values and $k/2$ regions with the largest response values
  end if
  Generate supporting features s_{di} for each region
end for
Maximize the likelihood according to Eq. (9) using stochastic gradient descent

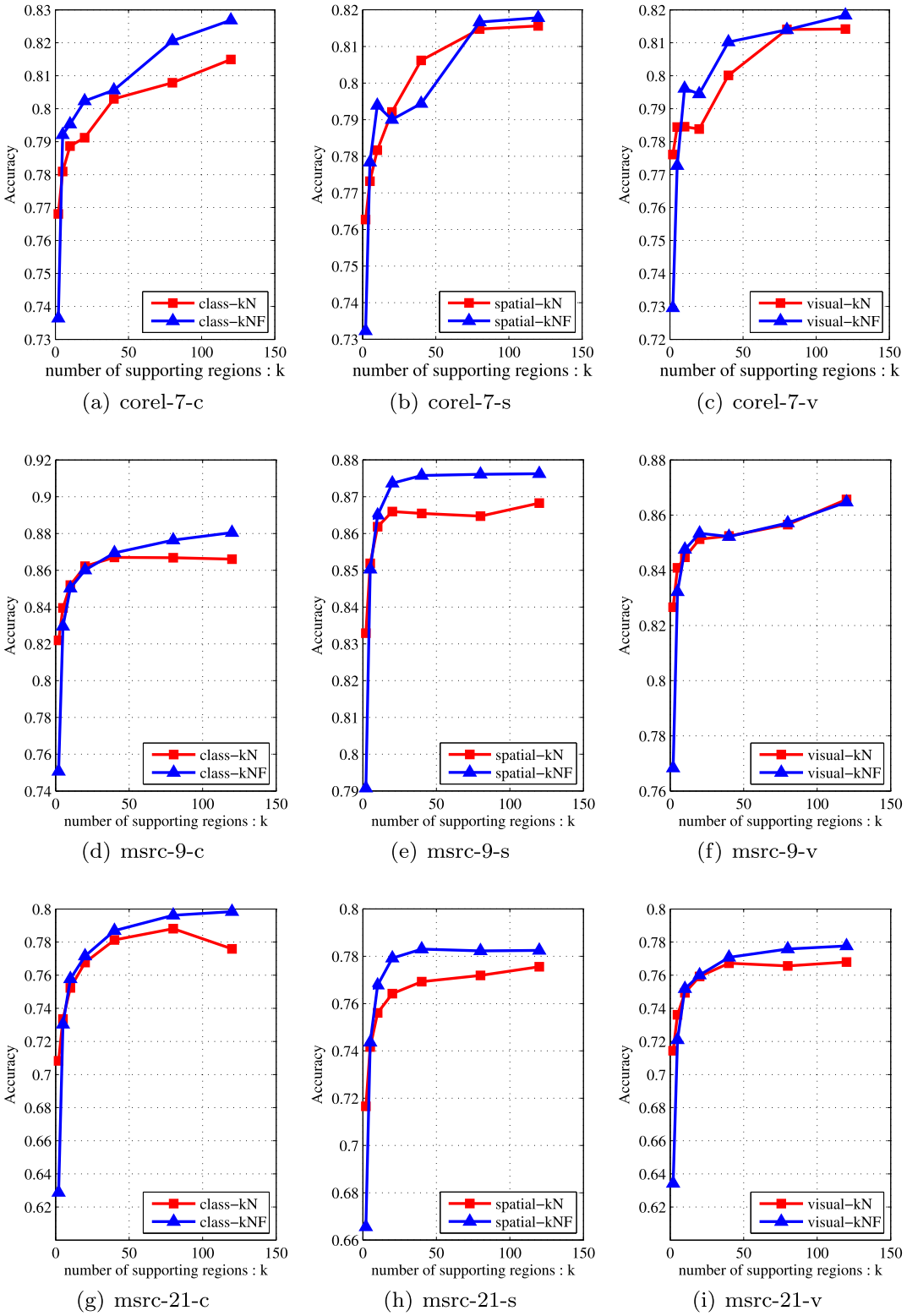


Fig. 6 Annotation accuracies of ASRG with different supporting region selection strategies. The *top row* shows the performance on the 7-class Corel dataset by selecting supporting regions with c (class), s (spatial) and v (visual). The *second row* shows the performance on the 9-class MSRC dataset and the *bottom row* shows experimental results on the 21-class MSRC dataset. The *curve* with square markers is the accuracy of using kN (k nearest) supporting regions and the *curve with triangle markers* is that of using kNF ($k/2$ nearest and $k/2$ farthest) supporting regions, where k takes 2, 5, 10, 20, 40, 80, 120

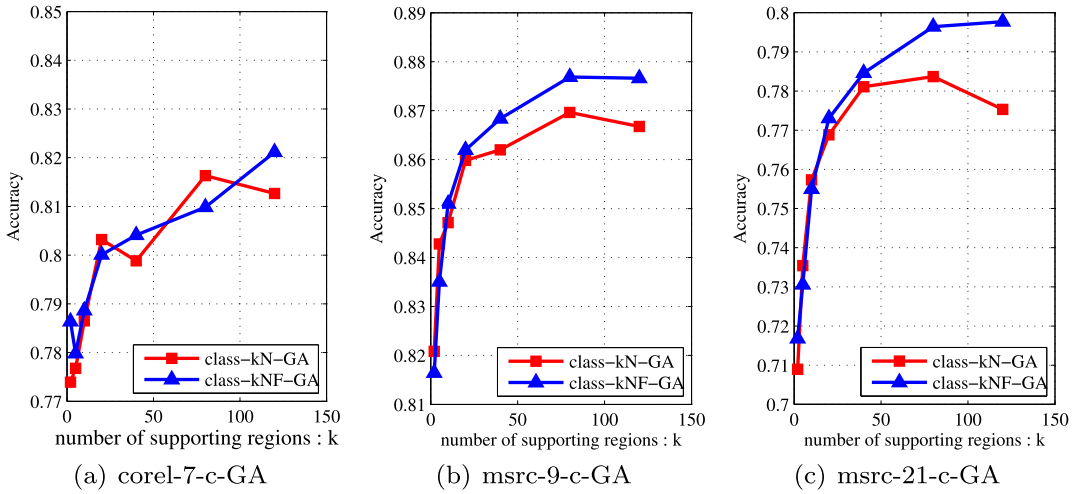


Fig. 7 Annotation accuracies of ASRG with supporting region selected by utilizing class response with optimized weights. The *curve with square markers* is the accuracy of using kN (k nearest) supporting regions and the *curve with triangle markers* is that of using kNF ($k/2$ nearest and $k/2$ farthest) supporting regions, where k takes 2, 5, 10, 20, 40, 80, 120

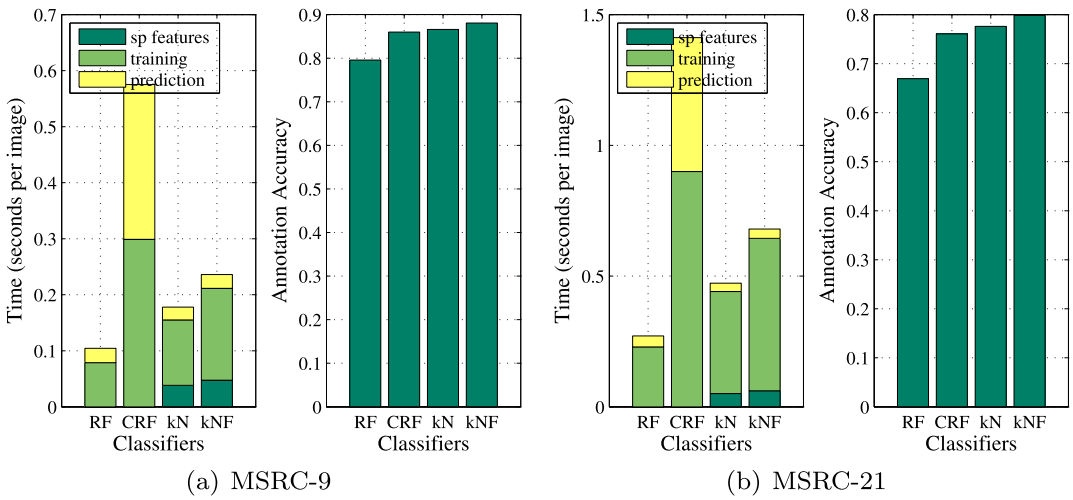


Fig. 8 Time costs of training, prediction and generating supporting features (sp features) and accuracy comparisons of RF, CRF and ASRG on the 9-class and 21-class MSRC dataset. In ASRG, supporting regions were selected by utilizing class-kN and class-kNF with $k = 120$

coordinates of image regions. Figure 6 shows the accuracy of ASRG with supporting regions selected using (1)–(6).

With the increase of k, the annotation accuracy has been significantly improved. The performance of kNF is better than kN on all three datasets because kNF not only uses contextual information from the selected regions with small distances, but also captures the co-occurrences of "foreground" and "background" regions. Besides, using class probabilities to select supporting regions achieves better performance than using spatial and visual features. Figure 7 shows the accuracy of ASRG with supporting regions selected using (7) and (8). By comparing Figs. 6 and 7, we observe that the optimized weights achieves better annotation accuracy when k is small. However, with the increase of k, directly using $w = 1$ may achieve similar performance to that of optimized weights. Therefore, we directly set $w = 1$ if we choose large k for efficiency consideration.

5.4 Evaluation of training and prediction speed

In order to evaluate the time cost of ASRG, we compared ASRG with two state-of-art classifiers, random forest (RF) and conditional random fields (CRF). ASRG is a light-weighted classifier, with which training and prediction is much efficient than CRF. In order to utilize ASRG for annotation, supporting regions were selected and supporting features were then calculated. Figure 8 shows the comparisons of time costs of RF, CRF and ASRG on two MSRC

datasets. In ASRG, supporting regions were selected by utilizing class-kN and class-kNF with $k = 120$. We observe that training time of ASRG is much less than CRF and slightly more than RF. The prediction speed of ASRG is also very fast and similar to RF. The reason is that prediction with CRF need to do iterative inference but prediction with ASRG is able to output the conditional probability directly. To summarize, ASRG is much efficient in both training and prediction procedure. Moreover, ASRG achieves better performance on three datasets compared with CRF.

Different with RF and CRF, ASRG needs extra time to generate supporting features. The time to generate supporting features of class-kNF is slightly more than that of class-kN, and the time of training ASRG is similar. Figure 9 shows time costs of training ASRG with different number of supporting regions selected using class probabilities on the 9-class MSRC dataset. With the increase of k, both the time of generating supporting features and training ASRG do not increase. It is due to the fact that no matter how many supporting regions we selected, the weighted distances of all region pairs were needed to be computed as described in Eq. (12). After that, regions in the image were sorted and k supporting regions were selected. Then, supporting features were generated from selected regions. As the main time costs came from calculating distances, k did not influence the time to generate supporting features. Since the dimension of supporting features was fixed and did not increase with k, the training time did not increase either.

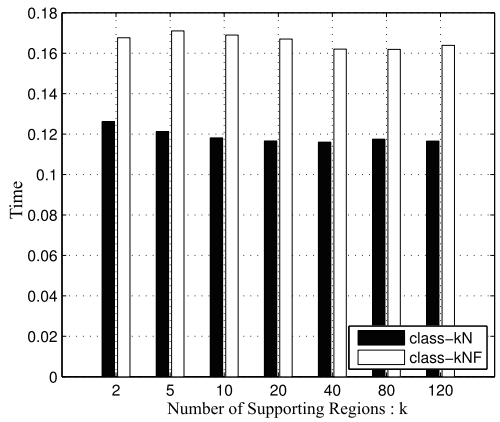


Fig. 9 Time costs (*seconds per image*) of training ASRG with different number of supporting regions selected using class probabilities on the 9-class MSRC dataset

5.5 Comparing annotation accuracy with the state-of-art methods

We used Random Forests (RF) and Conditional Random Fields (CRF) as baseline methods. As shown in Fig. 6, ASRG with supporting regions selected using class probabilities achieves better performance than using spatial and visual features. Meanwhile, as described above, directly using $w = 1$ achieves similar performance to that of using optimized weights while being more efficient. Thus, we only evaluated the performance of ASRG with supporting regions selected using class probabilities and $w = 1$. Tables 3, 4 and 5 provides the comparisons of annotation accuracy on each class and the overall accuracy on different methods on all three datasets. We compared four different algorithms, RF, CRF, ASRG with class-120N and ASRG with class-120NF. On all three datasets, ASRG performs better than RF and CRF, in which ASRG with class-120NF is further better than ASRG with class-120N.

Table 6 shows the performance comparisons of the proposed methods and other state-of-art methods. We evaluated the performance of ASRG with 120 supporting regions selected using class, visual and spatial. Selecting supporting regions with class-120NF achieves the best performance on

Table 3 The annotation accuracies of different methods on 7-class Corel dataset using RF, CRF, ASRG with class-120N supporting regions and ASRG with class-120NF supporting regions

Algorithm	Rhino	Polar bear	Water	Snow	Vegetation	Ground	Sky	Overall
RF	73	79	68	82	67	78	**81**	74
CRF	79	**85**	78	90	72	85	**81**	81
c-120N	79	80	**87**	90	71	84	**81**	81
c-120NF	**80**	82	85	**91**	**73**	**87**	**81**	**83**

Table 4 The annotation accuracies of different methods on 9-class MSRC dataset using RF, CRF, ASRG with class-120N supporting regions and ASRG with class-120NF supporting regions

Algorithm	Building	Grass	Tree	Cow	Sky	Airplane	Face	Car	Bicycle	Overall
RF	74	**95**	80	58	93	47	76	66	62	80
CRF	**84**	94	**87**	77	95	67	80	79	72	86
c-120N	82	95	84	76	95	69	**95**	85	71	87
c-120NF	81	95	86	**80**	**96**	**78**	92	**87**	**74**	**88**

Table 5 The annotation accuracies of different methods on 21-class MSRC dataset using RF, CRF, ASRG with class-120N supporting regions and ASRG with class-120NF supporting regions

Algorithm	Building	Grass	Tree	Cow	Sheep	Sky	Airplane	Water	Face	Car	Bicycle
RF	64	**94**	78	42	59	91	32	65	59	44	51
CRF	**82**	92	**87**	61	82	93	59	**73**	63	72	66
ASRG c-120N	74	94	83	66	**83**	**95**	61	63	68	**79**	69
ASRG c-120NF	79	93	85	**74**	81	**95**	**74**	68	**75**	**79**	**74**

Algorithm	Flower	Sign	Bird	Book	Chair	Road	Cat	Dog	Body	Boat	Overall
RF	62	29	3	67	11	71	37	22	38	3	67
CRF	79	**51**	6	83	24	73	57	35	54	4	76
ASRG c-120N	**92**	46	16	**89**	**47**	79	46	49	59	**25**	78
ASRG c-120NF	81	49	**27**	**89**	42	**82**	**72**	**54**	**68**	**25**	**80**

Table 6 Comparison of our results on three datasets with other state-of-art methods

Algorithm	MSRC-9	MSRC-21	Corel-7
Multiscale CRF [14]	–	–	**80.0 %**
PLSA-MRF [36]	82.3 %	73.5 %	–
Textonboost [30]	–	72.7 %	74.6 %
CRFσ loc + glo [37]	84.9 %	–	–
TextonForests [29]	–	72 %	–
Schroff et al. [27]	87.2 %	71.7 %	–
Yang et al. [39]	–	75.1 %	–
Gould et al. 08 [12]	**88.5 %**	76.5 %	77.3 %
Gould et al. 09 [11]	–	76 %	–
Munoz et al.[23]	–	78 %	–
Harmony Potential [10]	–	77 %	–
Hierarchical CRF [18]	–	**86 %**	–
ASRG c-120N	86.6 %	77.6 %	81.5 %
ASRG c-120NF	**88.1 %**	**79.8 %**	**82.7 %**
ASRG s-120N	86.8 %	77.6 %	81.6 %
ASRG s-120NF	87.6 %	78.3 %	81.8 %
ASRG v-120N	86.6 %	76.8 %	81.4 %
ASRG v-120NF	86.5 %	77.8 %	81.8 %
EASRG	87.9 %	77.1 %	82.1 %

the Corel and the 21-class MSRC dataset. On the 9-class MSRC dataset, it achieves similar performance comparing with [12] and is better than other methods. On the 21-class MSRC dataset, Ladicky et al. [18] achieved 86 % on this dataset. However, they achieved 81 % by just using baseline CRF, which is better than other methods, and this is probably due to its using of complex visual features. In [18], the features of each pixel was extracted; but in our method visual features were extracted from the segmented regions, which was much faster. We also evaluated the performance of EASRG, which used all regions in the image as supporting regions. The experimental results show that EASRG achieves competitive performance compared with other methods. Since EASRG does not need to select supporting regions, it directly generates supporting features from all regions in the image, which can be regarded as a type of global features. Thus, it is a bit faster than ASRG which comes with selection procedures. Figure 10 illustrates

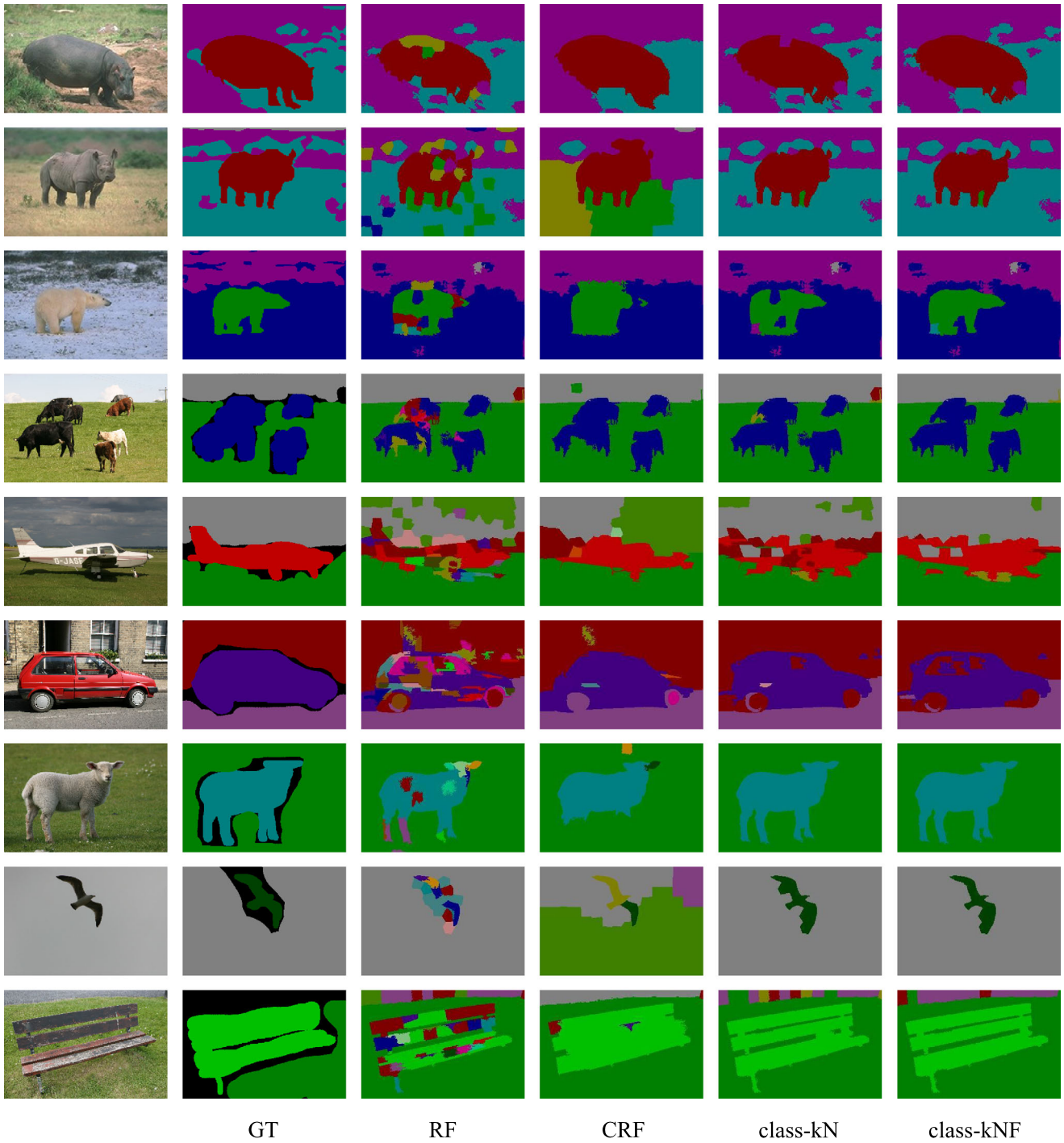


Fig. 10 Annotation results of the sample images from three datasets. The *first* and the *second columns* show the sample images and their ground-truth annotations (GT). The *third column* shows the annotation results of Random Forests (RF), and the *fourth column* shows the annotation results of Conditional Random Fields (CRF). The *right two column* show the annotation results of ASRG with supporting regions selected using class-kN and class-kNF, with $k = 120$

the exemplar images, ground-truth annotations and the annotation results using different methods.

The proposed ASRG model is also capable of utilizing supporting regions selected from different images. On the 21-class MSRC dataset, we manually selected two similar training images for each testing image, and for each region we selected supporting regions from the image and other two selected images, by which the overall accuracy achieves 83 %. In our future work, we intend to work on selecting supporting regions in similar images automatically. Mean-

while, this can also be employed to process videos by selecting supporting regions between different frames.

6 Conclusion

This paper proposes a directed graph model aiming to capture contextual information extracted from selected surrounding regions. Improved on the traditional context-based classification which utilized adjacent regions as supporting regions, such as Conditional Random Fields, we use the supporting regions selected from surrounding image regions by using the class labels, visual or spatial information in the proposed new graph model and achieve better performance than that of CRF and other state-of-art methods.

Acknowledgements This work is supported by the Program for New Century Excellent Talents of Ministry of Education, China (Grant No. NCET-11-0213), the National Natural Science Foundation of China (Grant Nos. 61273257, 61035003, 61021062), the "863" Program of China (Grant No. 2011AA01A202), and the "973" Program of China (Grant No. 2010CB327903).

References

1. Achanta R, Shaji A, Smith K, Lucchi A, Fua P, Süsstrunk S (2010) Slic superpixels. EPFL Technical Report 149300
2. Achanta R, Shaji A, Smith K, Lucchi A, Fua P, Süsstrunk S (2012) Slic superpixels compared to state-of-the-art superpixel methods. IEEE Trans Pattern Anal Mach Intell 34(11):2274–2282
3. Andreetto M, Zelnik-Manor L, Perona P (2012) Unsupervised learning of categorical segments in image collections. IEEE Trans Pattern Anal Mach Intell 34(9):1842–1855
4. Bosch A, Zisserman A, Muoz X (2007) Image classification using random forests and ferns. In: Proceedings IEEE international conference on of computer vision, pp 1–8
5. Cao L, Fei-Fei L (2007) Spatially coherent latent topic model for concurrent segmentation and classification of objects and scenes. In: Proceedings of IEEE international conference on computer vision, pp 1–8
6. Chatterjee S (2013) Vision-based rock-type classification of limestone using multi-class support vector machine. Appl Intell, 1–14
7. Criminisi A (2004) Microsoft research Cambridge object recognition image database. http://research.microsoft.com/vision/cambridge/recognition/
8. Escalante H, Montes M, Sucar LE (2007) Improving automatic image annotation based on word co-occurrence. In: Proceedings of adaptive multimedial retrieval: retrieval, user, and semantics, pp 57–70
9. Fulkerson B, Vedaldi A, Soatto S (2009) Class segmentation and object localization with superpixel neighborhoods. In: Proceedings of IEEE international conference on computer vision, pp 670–677
10. Gonfaus J, Boix X, Van De Weijer J, Bagdanov A, Serrat J, Gonzalez J (2010) Harmony potentials for joint classification and segmentation. In: Proceedings of IEEE conference on computer vision and pattern recognition, pp 3280–3287
11. Gould S, Fulton R, Koller D (2009) Decomposing a scene into geometric and semantically consistent regions. In: Proceedings of IEEE international conference on computer vision, pp 1–8
12. Gould S, Rodgers J, Cohen D, Elidan G, Koller D (2008) Multi-class segmentation with relative location prior. Int J Comput Vis 80(3):300–316
13. Gould S, Russakovsky O, Goodfellow I, Baumstarck P, Ng A, Koller D (2010) The stair vision library (v2.4). http://ai.stanford.edu/~sgould/svl
14. He X, Zemel R, Carreira-Perpinan M (2004) Multiscale conditional random fields for image labeling. In: Proceedings of IEEE conference on computer vision and pattern recognition, vol 2, pp 695–702
15. Heckerman D, Chickering D, Meek C, Rounthwaite R, Kadie C (2001) Dependency networks for inference, collaborative filtering, and data visualization. J Mach Learn Res 1:49–75
16. Hossain MJ, Dewan MAA, Chae O (2012) A flexible edge matching technique for object detection in dynamic environment. Appl Artif Intell 36(3):638–648
17. Kumar S, Hebert M (2005) A hierarchical field framework for unified context-based classification. In: Proceedings of the tenth IEEE international conference on computer vision, pp 1284–1291
18. Ladicky L, Russell C, Kohli P, Torr P (2009) Associative hierarchical crfs for object class image segmentation. In: Proceedings of IEEE international conference on computer vision, pp 739–746
19. Lee S, Le HX, Ngo HQ, Kim HI, Han M, Lee YK et al (2011) Semi–Markov conditional random fields for accelerometer-based activity recognition. Appl Artif Intell 35(2):226–241
20. Lim J, Arbeláez P, Gu C, Malik J (2009) Context by region ancestry. In: Proceedings of IEEE international conference on computer vision, pp 1978–1985
21. Liu W, Yang Y (2009) Structural context for object categorization. In: Proceedings of pacific rim conference on multimedia: advances in multimedia information processing, pp 280–291
22. Mirghasemi S, Yazdi HS, Lotfizad M (2012) A target-based color space for sea target detection. Appl Artif Intell 36(4):960–978
23. Munoz D, Bagnell J, Hebert M (2010) Stacked hierarchical labeling. In: Proceedings of European conference on computer vision, pp 57–70
24. Nebti S, Boukerram A (2013) Handwritten characters recognition based on nature-inspired computing and neuro-evolution. Appl Artif Intell 38(2):146–159
25. Oliva A, Torralba A et al (2007) The role of context in object recognition. Trends Cogn Sci 11(12):520–527
26. Rabinovich A, Vedaldi A, Galleguillos C, Wiewiora E, Belongie S (2007) Objects in context. In: Proceedings of IEEE international conference on computer vision, pp 1–8
27. Schroff F, Criminisi A, Zisserman A (2008) Object class segmentation using random forests. In: Proceedings of the British machine vision conference, pp 1–10
28. Shi Y, Gao Y, Wang R, Zhang Y, Wang D (2013) Transductive cost-sensitive lung cancer image classification. Appl Artif Intell 38(1):16–28
29. Shotton J, Johnson M, Cipolla R (2008) Semantic texton forests for image categorization and segmentation. In: Proceedings of IEEE conference on computer vision and pattern recognition, pp 1–8
30. Shotton J, Winn J, Rother C, Criminisi A (2006) Textonboost: joint appearance, shape and context modeling for multi-class object recognition and segmentation. In: Proceedings of European conference on computer vision, pp 1–15
31. Ting CY, Phon-Amnuaisuk S (2010) Optimal dynamic decision network model for scientific inquiry learning environment. Appl Artif Intell 33(3):387–406
32. Torralba A (2003) Contextual priming for object detection. Int J Comput Vis 53(2):169–191
33. Tu Z (2008) Auto-context and its application to high-level vision tasks. In: Proceedings of IEEE conference on computer vision and pattern recognition, pp 1–8

34. Uddin MZ, Lee J, Kim TS (2010) Independent shape component-based human activity recognition via hidden Markov model. Appl Artif Intell 33(2):193–206
35. Valova I, Milano G, Bowen K, Gueorguieva N (2011) Bridging the fuzzy, neural and evolutionary paradigms for automatic target recognition. Appl Artif Intell 35(2):211–225
36. Verbeek J, Triggs B (2007) Region classification with Markov field aspect models. In: Proceedings of IEEE conference on computer vision and pattern recognition, pp 1–8
37. Verbeek J, Triggs W et al (2008) Scene segmentation with crfs learned from partially labeled images. In: Advances in neural information processing systems, vol 20, pp 1553–1560
38. Wolf L, Bileschi S (2006) A critical view of context. Int J Comput Vis 69(2):251–261
39. Yang L, Meer P, Foran D (2007) Multiple class segmentation using a unified framework over mean-shift patches. In: Proceedings of IEEE conference on computer vision and pattern recognition, pp 1–8
40. Yang YB, Li YN, Pan LY, Li N, He GN (2013) Image retrieval based on augmented relational graph representation. Appl Intell, 1–13

Qiao-Jin Guo received the B.Sc. degree in computer science from Nanjing University, Nanjing, China, in 2007. He is currently a Ph.d. student at the Department of Computer Science of Nanjing University. His current research interests include image annotation and machine learning.

Ning Li received the Ph.D. degree in computer science from Nanjing University, Nanjing, China, in 2012. She is currently an Associate Professor with the Department of Computer Science of Nanjing University. Her current research interests focus on machine learning and semantic based image retrieval.

Yu-Bin Yang received the B.Sc. degree in computer science from Wuhan Technical University of Surveying and Mapping, Wuhan, China, in 1997, and the M.Sc. and Ph.D. degrees in computer science from Nanjing University, Nanjing, China, in 2000 and 2003, respectively. He participated in collaborative research at The Chinese University of Hong Kong (CUHK) in 2003–2005, and at University of New South Wales at Australian Defence Force Academy (UNSW@ADFA) in 2005–2006. He is currently an Associate Professor with the State Key Laboratory for Novel Software Technology, and the Department of Computer Science of Nanjing University. His current research interests include semantic-based media computing, multimedia information retrieval, large-scale data mining, and machine learning.

Gang-Shan Wu is currently a professor of Department of Computer Science and Technology at Nanjing University China. He got his Ph.D., M.S. and B.Sc. from Department of Computer Science and Technology at Nanjing University in 2000, 1991 and 1988 respectively. His current research interests include multimedia content analysis, multimedia information retrieval and digital museum.