Feature Voting for Object Localization via Density Ratio Estimation

Liantao Wang[1*], Dong Deng[1] and Chunlei Chen[2]
[1]College of Internet of Things Engineering, Hohai University
Changzhou, Jiangsu 213022 - China
[e-mail: ltwang@hhu.edu.cn]
[2]School of Computer Engineering, Weifang University
Weifang, Shandong 261061 – China
[e-mail: chunlei.chen@wfu.edu.cn]
*Corresponding author: Liantao Wang

Received January 1, 2019; revised March 3, 2019; revised April 27, 2019; accepted June 6, 2019; published December 31, 2019

Abstract

Support vector machine (SVM) classifiers have been widely used for object detection. These methods usually locate the object by finding the region with maximal score in an image. With bag-of-features representation, the SVM score of an image region can be written as the sum of its inside feature-weights. As a result, the searching process can be executed efficiently by using strategies such as branch-and-bound. However, the feature-weight derived by optimizing region classification cannot really reveal the category knowledge of a feature-point, which could cause bad localization. In this paper, we represent a region in an image by a collection of local feature-points and determine the object by the region with the maximum posterior probability of belonging to the object class. Based on the Bayes' theorem and Naive-Bayes assumptions, the posterior probability is reformulated as the sum of feature-scores. The feature-score is manifested in the form of the logarithm of a probability ratio. Instead of estimating the numerator and denominator probabilities separately, we readily employ the density ratio estimation techniques directly, and overcome the above limitation. Experiments on a car dataset and PASCAL VOC 2007 dataset validated the effectiveness of our method compared to the baselines. In addition, the performance can be further improved by taking advantage of the recently developed deep convolutional neural network features.

Keywords: Naive-Bayes, feature-voting, object localization, density ratio estimation, feature-scoring

This work was supported in part by the National Natural Science Foundation of China (NSFC) (No. 61703139).

http://dx.doi.org/10.3837/tiis.2019.12.012 **ISSN : 1976-7277**

1. Introduction

Since efficient subwindow search [1] was first proposed to supplant the time-consuming sliding window for object localization, many extensions [2, 3, 4, 5, 6, 7, 8] of it have emerged and constituted an important branch in the literature. All of these methods can be unified in a feature-voting framework that consists of two components: evaluation function that measures how likely a region belonging to the object class and that is additive in terms of feature-points (i.e., grid SIFT), and search strategy that can rapidly determine the region with the maximum score in the image space.

Many of the extensions [2, 5, 6, 8] adopt linear SVM with bag-of-features representation to provide an additive classifier, and decompose the SVM score of a region into the sum of feature-weights to facilitate the usage of efficient search strategies. However the scheme does not work well in object localization because 1) the feature-weights optimized for classification at a region-level cannot perform well for feature-point classification; 2) important information is lost during quantization for bag-of-features representation.

The methods based on a SVM classification system may have drawbacks when shifted to an object localization task, since a region with maximum classification score does not necessarily correspond to the best result for object location. When the feature-weights are positively biased, the search result tends to cover much background, sometimes even the whole image; On the contrary, when the feature-weights are negatively biased, the search result would be a tiny region covering a few feature-points, leading to incomplete localization. For some of the methods based on SVM classification, the feature-weights are heuristically tuned with a little fraction of the bias item in their implementation s to generate good results [5].

The quantization loss is also criticized in image classification [9]. The authors instead propose Naive-Bayes Nearest Neighbour (NBNN) to compute the Image-to-Class distance in the space of feature-points without quantization. The NBNN motivates several variants [10, 11, 12]. Inspired by the success of the NBNN in image classification, researchers apply it to object localization [10, 4, 7]. These methods approximate the feature density with nearest neighbour distance and use a feature voting scheme for object localization.

In this paper, we enrich the feature-voting framework with a new formulation for object localization. Note that we do not intend to propose new efficient search methods. Hence it is not about improving object localization efficiency. Rather the contribution is an evaluation function and a novel feature scoring strategy that aim to overcome the limitations mentioned above. It can work with any previous search strategies such as branch-and-bound [1] and branch-and-cut [5]. Given a test image, we intend to locate the object by the region that has the maximum posterior probability of belonging to the object class. In order to formulate this intuition, we represent a region by a collection of local features such as SIFT [13] and SURF [14] for images, STIP [15] for videos, and deep pixel features [16, 17]. We further demonstrate that the posterior probability of a region can be reformulated as the sum of feature-scores based on Bayes' theorem and Naive-Bayes assumption.

Although the methods in [10, 4, 7], which can be categorized into the feature-voting framework, obtain similar ratio form of feature score, they use the nearest neighbor distance to approximate kernel density estimation to estimate two densities separately. Compared to computing two densities independently and then taking the ratio, the superiority of estimating the ratio directly has been demonstrated thoroughly [18]. We therefore model the feature score

with Gaussian kernels, and generate more accurate estimation for the likelihood of a feature-point belonging to the object class.

Our contributions in this paper include the following three aspects. 1) We analyze the object localization methods based on SVM with bag-of-features, which constitute an important branch in object localization field, and point out their drawback. 2) We enrich the feature-voting framework for object localization by a maximum posterior scheme. 3) We model the feature score with Gaussian kernels, which gains more benefits from generative model compared to the nearest neighbour approximations. 4) The recently developed deep convolutional local features can be adopted and greatly improve the performance.

The remainder of this paper is organized as follows: We unify related works in a feature-voting framework, and provide a relatively comprehensive literature review in Section 2. In Section 3, we analyze the object localization methods based on SVM classification, and point out their drawback and the cause. Then the new formulation for object localization and the feature-score estimation are detailed in Section 4 and 5 respectively. Experimental results are reported in Section 6. Section 7 concludes this work.

2. Literature Review

2.1 Feature Voting for Object Localization

Many methods for efficient object localization can be unified in a feature-voting framework. In such a framework, an evaluation function $f:\mathcal{X}\longrightarrow\mathbb{R}$ measures a region $R\in\mathcal{X}$ how likely it belongs to the object class. Consequently the object localization in a new image I is to find a region maximizing the score: $R^{*}=\arg\max_{R\in I} f(R)$. This evaluation function is required to be additive in terms of feature-points to facilitate the usage of efficient search strategies: $f(R)=\sum_{x_i\in R} f(x_i)$, where x_i represents a local feature-point in the region R. We next review the related works according to these two components.

1) Evaluation Function. For methods [2, 1, 5, 6, 8] based on SVM classification, the evaluation function is just the SVM decision function. In order to ensure the additivity, the kernel is restricted to linear and the feature is expressed as bag-of-words histograms. Thus the score of a region can be decomposed into the sum of the feature-scores, which are actually the weights of the visual words the feature-points mapping to. Lehmann et al. [3] use Footprint [19] to score a region, and express the score as the sum of each feature contribution. Boiman et al. [9] propose image-to-class distance for classification. Based on the same Naive-Byaes assumption, object localization is formulated as a feature voting problem from the perspective of mutual information and likelihood respectively [10, 4, 7]. They all employ nearest neighbour distance to approximate a feature score. [7] and [10] contribute different strategies to adapt kernel bandwith, while [4] propose to merely use the positive features to localize objects.

2) Efficient Search Strategy. The original search strategy proposed in ESS can be directly applied to videos, but it is too slow due to the dimension increase. In order to speed up the search process, Yuan et al. [7, 20] split the search space into two subspaces and propose a tighter upper bound for subvolume search in videos. However, the localization form of these methods is limited to rectangles for images or cubes for videos, and it will be rather restrictive for non-boxy objects (dogs, snakes, etc.). In order to address this limitation, a more flexible

localization form [6] using composite bounding boxes and bounding polygons is proposed under the branch-and-bound framework. Fundamentally, Vijayanarasimhan and Grauman [5] introduce a branch-and-cut approach to determine the best-scoring region that can be arbitrary shape. In [5], an image is first over-segmented into super-pixels, and a region-graph is constructed according to the super-pixel adjacency. Then the object search is formulated as finding the maximum-weight connected subgraph, and solved as a prize-collecting Steiner tree problem [21]. The branch-and-cut is also exploited with more accurate pre-segmentation [8], and applied to action localization [2].

2.2 Density Ratio Estimation

New statistical data processing towards the ratio of probability densities has been developed in machine learning community [18]. A naive approach to estimating a density ratio is to estimate two densities separately and then take their ratio. However, this naive approach is not reliable especially in high-dimensional problems [18]. To overcome this issue, various approaches have been proposed to estimate density ratio without going through density estimation [22, 23, 24, 25, 26]. Density ratio estimation has been used for non-stationary adaptation [27], outlier detection [28] and change-point detection [29, 30]. Doersch et al. [31] propose a discriminative variant of mean-shift for finding local maxima of density ratios that correspond to discriminant patches. Wang et al. [32] employ least squares density ratio estimation to compute feature score and search maximum-scored connected-regions.

2.3 Deep Local Features

Recently, classical local feature extractors (e.g., SIFT [13]) have been substantially outperformed by the introduction of the latest generation of Convolutional Neural Networks (CNNs) [33]. In a typical CNN, the last layer captures more discriminant semantic information that is significant for classification, while the earlier layers contains more spatial information which serves as an important factor for localization. To get the best of both worlds, some researchers have proposed concatenating outputs of multiple layers at a location to form a deep local feature representations for visual tracking [17] and objet localization [16]. Long et al. [34] introduce skip-connections and add high-level prediction layers to intermediate layers to generate pixel-wise prediction results at multiple resolutions. Badrinarayanan et al. [35] employ a convolutional encoder-decoder network with pooling index guided de-convolution modules to exploit the features from multi-level convolutional layers. Zhang et al. [36] propose a generic aggregating multi-level convolutional feature framework for salient object detection. It integrates multi-level feature maps into multiple resolutions, and learn to combine feature maps.

3. Drawback of Object Localization via SVM Classification

The object localization methods based on SVM classification depend on a crafty decomposition of the classifier score. Concretely, local descriptors (e.g. SIFT) are extracted densely for each image. Then k-means is applied to a sample of randomly selected training descriptors to get K cluster centers, which are actually used as visual vocabulary to quantize all the descriptors. For an image region R containing M local feature-points $\{x_1, x_2, \cdots, x_M\}$, where $x_i \in \mathbb{R}^K$ indicates the descriptor of the i-th local point, each descriptor x_i is mapped to its closest visual word which is indexed by $k_i \in \{1, 2, \ldots, K\}$. Thus

the image region can be described as a histogram $h(R)$, by counting the number of feature-points mapped to every visual word.

With the histogram feature representation, N images of training data containing objects of interest or not can be expressed as $T=\{(h(I_1),y_1),\cdots,(h(I_N),y_N)\}$, where $h(I_i)\in\mathbb{R}^K,\forall i$ and $y_i\in\{-1,1\},\forall i$. We can obtain a linear SVM classifier by optimizing:

$$\begin{aligned}\min_{w,b,\xi}\quad & \frac{1}{2}\|w\|_2^2+C\sum_{i=1}^{N}\xi_i\\ \text{s.t.}\quad & y_i(w^{\mathrm{T}}h(I_i)+b)\geqslant 1-\xi_i,\ \forall i;\\ & \xi_i\geqslant 0,\ \forall i.\end{aligned}\tag{1}$$

Here, w is the weight imposed on different feature dimensions; ξ is the slack variable which allows for penalized constraint violation; C is the parameter controlling the trade-off between having a larger margin and violating less constraints; b is the bias item. (1) is usually solved with the dual problem:

$$\begin{aligned}\min_{\alpha}\quad & \frac{1}{2}\sum_{i,j=1}^{N}\alpha_i\alpha_j y_i y_j(h(I_i)\cdot h(I_j))-\sum_{i=1}^{N}\alpha_i\\ \text{s.t.}\quad & \sum_{i=1}^{N}\alpha_i y_i=0;\\ & 0\leq\alpha_i\leq C,\forall i.\end{aligned}\tag{2}$$

where α is the Lagrange multiplier. (2) can be solved by a typical programming package. Then for a region R in a test image, its decision value can be calculated by:

$$f(R)=\sum_{i=1}^{N}y_i\alpha_i\left(h(I_i)\cdot h(R)\right)+b.\tag{3}$$

Let h^j denote the j-th bin of the histogram h. It is associated with the feature-weight $w_j=\sum_{i=1}^{N}y_i\alpha_i h^j(I_i)$, and the decision value can be reformulated as:

$$f(R)=\sum_{j=1}^{K}w_j h^j(R)+b=\sum_{i=1}^{M}w_{k_i}+b,\tag{4}$$

where k_i is the index of the visual word that feature-point x_i is mapped to.

The bias item b can be omitted for maximizing $f(R)$. Thus the score of the region R is decomposed into the sum of its M feature-weights, and the object search problem can be formulated as:

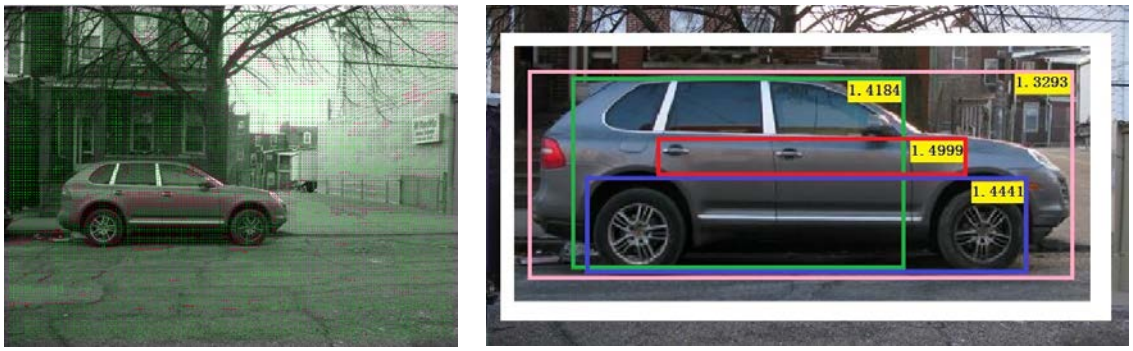
$$R^*=\max_{R\subseteq I}\sum_{i=1}^{M}w_{k_i},\tag{5}$$

where w_{k_i} is the SVM weight for k_i-th feature. w_{k_i} is back projected to the local feature-point x_i through the visual vocabulary and can be considered as the score of the feature-point that contributes to the object localization. That is why we call it a feature-voting

framework. The region search process can be conducted efficiently by ESS [1] or ERS [5] to generate sub-window and irregular localization respectively.

It is an ingenious formulation deduction that transforms an image region score into the summation of its inside feature-points, but the object search result could be awful. The fundamental cause is that SVM aims to find the maximum margin to separate the regions in an image into two categories, and that does not guarantee the region with the maximum response is the perfect object location. From another perspective, the feature-weights derived by minimizing the generalization error of a region classification cannot reflect the category of the feature-points, i.e., a feature-point with negative weight does not mean it belongs to non-object, but it will contribute negatively in the search process mentioned above and tend to be excluded from the object location.

We present an actual example in **Fig. 1**. The feature-weights are plotted in (a) for the learned SVM with $b = 1.4848$, where a red point indicates a positive weight, and green negative, and brighter color means higher absolute value. In (b) we plot some of the positive sub-windows according to the SVM classification, with the decision score shown in the upper right corner. For the sake of clarity, we enlarge the local region around the car and use different colors to distinguish sub-windows. It can be seen that although the SVM classifier does well in the region classification of car and non-car, the feature-weights do not have reliable clues for the categories of the feature-points: many of the feature-points on the car are assigned negative weights, and some in the background are assigned with positive ones. This consequently leads to a bad localization result: the region with the maximum score (1.4999) is the sub-window only covering a fraction of the car.



(a) Feature-weights (b) Positive subwindows

Fig. 1. An actual example of the object localization based on SVM classification. The subwindow with the maximal score does not correspond to the one that can perfectly fit the object.

4. Our Method

In this section, we aim to address the problem of applying SVM classifier to object localization. We follow the feature-voting framework, which consists of an evaluation function additive in terms of feature-scores and an efficient search strategy. We adopt the posterior probability of belonging to the object as the evaluation function, and derive a feature-score in the form of density ratio based on Bayes' theorem and Naive-Bayes assumption. As for the feature-score estimation, we model the density ratio with Gaussian kernels, and solve the model by minimizing the KL-divergence of a probability density and its estimation among the training feature-points, that has been studied in [37].

4.1 Object Localization via Maximum Posterior

In order to determine the object location in an image I, we can find a region R^* that maximizes its posterior probability of belonging to the object class:

$$R^* = \arg\max_{R \subseteq I} p(y=1 \mid R). \tag{6}$$

The posterior probability can be rewritten with Bayes' theorem as:

$$\begin{aligned} & p(y=1 \mid R) \\ &= \frac{p(R \mid y=1)\,p(y=1)}{p(R \mid y=1)\,p(y=1) + p(R \mid y=-1)\,p(y=-1)} \\ &= 1 \Big/ \left(1 + \frac{p(R \mid y=-1)}{p(R \mid y=1)} \cdot \frac{p(y=-1)}{p(y=1)}\right). \end{aligned} \tag{7}$$

Assume the ratio of prior probabilities of object and background is a constant, then maximizing (7) can be equivalently transformed to

$$\max_{R \subseteq I} \frac{p(R \mid y=1)}{p(R \mid y=-1)}. \tag{8}$$

Based on the Naive-Bayes assumption that the local feature-points are i.i.d given its class, it can be rewritten as

$$\max_{R \subseteq I} \prod_{x_i \in R} \frac{p(x_i \mid y=1)}{p(x_i \mid y=-1)}. \tag{9}$$

Taking the logarithm, (9) becomes

$$\max_{R \subseteq I} \sum_{x_i \in R} \log \frac{p(x_i \mid y=1)}{p(x_i \mid y=-1)}. \tag{10}$$

We denote

$$f(x) = \log \frac{p(x \mid y=1)}{p(x \mid y=-1)} \tag{11}$$

as the score of feature-point x belonging to the object class. Then the object localization task is formulated as looking for a region that has a maximal summation of feature-scores, which can be achieved by readily employing branch-and-bound or branch-and-cut strategies. For a feature-point, if the object class likelihood is higher than the non-object class likelihood, the value of the feature-score is positive, and the point will contribute positively for the object. Otherwise, if the value of the feature-score is negative, the point will contribute negatively for the object. For simplicity of notation, we denote the class conditional probability densities by $p_+(x)$ and $p_-(x)$ respectively, and (11) becomes:

$$f(x) = \log \frac{p_+(x)}{p_-(x)}. \tag{12}$$

4.2 Feature Score Estimation

In this section, we model the density ratio $\gamma(x) = p_+(x) / p_-(x)$ with Gaussian kernels, and solve the model by minimizing the KL-divergence of a probability density and its estimation among the training feature-points, that has been studied in [37].

We model $\gamma(x)$ using a linear combination of Gaussian kernels:

$$\widehat{\gamma(x)} = w^{\top}\phi(x), \tag{13}$$

where $w = (w_1, w_2, \ldots, w_m)^{\top}$ is the parameter to be learned and m is the order of the model, $\phi(x) = \left(\phi_1(x), \phi_2(x), \ldots, \phi_m(x)\right)^{\top}$ is a set of Gaussian kernel functions.

We apply clustering to the training feature-points to get m centers $\{c_1, c_2, \ldots, c_m\}$, and use them to center the Gaussians. Therefore the kernel function takes the form of:

$$\phi_i(x \mid c_i, \sigma) = \exp\left(-\frac{\| x - c_i \|^2}{2\sigma^2}\right), \tag{14}$$

where σ is set as the average Euclidean length of the local descriptors.

In order to solve the above model, we minimize the KL-divergence of the density $p_+(x)$ and its estimation $\widehat{p_+(x)} = \widehat{\gamma(x)} p_-(x)$:

$$\begin{aligned} KL[p_+(x) \| \widehat{p_+(x)}] &= \int p_+(x) \log \frac{p_+(x)}{\widehat{\gamma(x)} p_-(x)} \mathrm{d}x \\ &= \int p_+(x) \log \frac{p_+(x)}{p_-(x)} \mathrm{d}x - \int p_+(x) \log \widehat{\gamma(x)} \mathrm{d}x \\ &= \int p_+(x) \log \frac{p_+(x)}{p_-(x)} \mathrm{d}x - \int p_+(x) \log\left(w^{\top} \phi(x)\right) \end{aligned} \tag{15}$$

Note that the first term in Eq. (15) is independent of w and can be ignored when the goal is to minimize the function with respect to w. Therefore the w can be solved by the following optimization:

$$\begin{aligned} \min_{w} \quad & -\mathbb{E}_+\left[\log\left(w^{\top}\phi(x)\right)\right] \\ \text{s.t.} \quad & \mathbb{E}_-\left[w^{\top}\phi(x)\right] = 1; \\ & w \geqslant 0. \end{aligned} \tag{16}$$

In the above optimization, the first constraint arises from the fact that $\hat{\gamma}$ is non-negative since it is a ratio of two probability densities; and the second constraint exists because $\widehat{p_+(x)}$ should be properly normalized to be a probability density.

Note that the expectations cannot be directly calculated since the probabilities are not accessible. We approximate them using sample means:

$$\mathbb{E}_+\left[\log\left(w^{\top}\phi(x)\right)\right] \approx \frac{1}{|\mathcal{I}_+|} \sum_{i \in \mathcal{I}_+} \log\left(w^{\top}\phi(x_i)\right), \tag{17}$$

$$\mathbb{E}_-\left[w^{\top}\phi(x)\right] \approx \frac{1}{|\mathcal{I}_-|} \sum_{i \in \mathcal{I}_-} w^{\top}\phi(x_i), \tag{18}$$

where we have denoted the indices of the feature-points from object class and background by $\mathcal{I}_+$ and $\mathcal{I}_-$ respectively.

Then the optimization becomes:

$$\begin{aligned} \min_{w} \quad & -\frac{1}{|\mathcal{I}_+|}\sum_{i\in\mathcal{I}_+}\log\left(w^\top\phi(x_i)\right) \\ \text{s.t.} \quad & \frac{1}{|\mathcal{I}_-|}\sum_{i\in\mathcal{I}_-}w^\top\phi(x_i)=1; \\ & w\succcurlyeq 0. \end{aligned} \tag{19}$$

This is a convex optimization problem and the global optimum can be obtained by a typical gradient descent method.

We summarize the proposed pipeline in **Fig. 2**. Given training images with bounding-box ground-truth, we first densely extract local features to obtain training feature-points. Then clustering is applied to the training feature-points to obtain m centers that are used to center the Gaussians in Eq. (14). After modeling the density ratio as a linear combination of the Gaussian kernels, we minimize (16) that is derived from KL-divergence. To this end, we need to use the sample means of the training feature-points to approximate the expectations therein, and obtain the new objective function (19). The solved parameter is used to calculate the density ratio (13). Once the density ratio estimation is completed, we can perform object localization for test images. We first densely extract feature-points and estimate their scores of (11) using (13), then search the maximum-score region using ESS and ERS to obtain bounding-box and irregular-shape localizations, respectively.

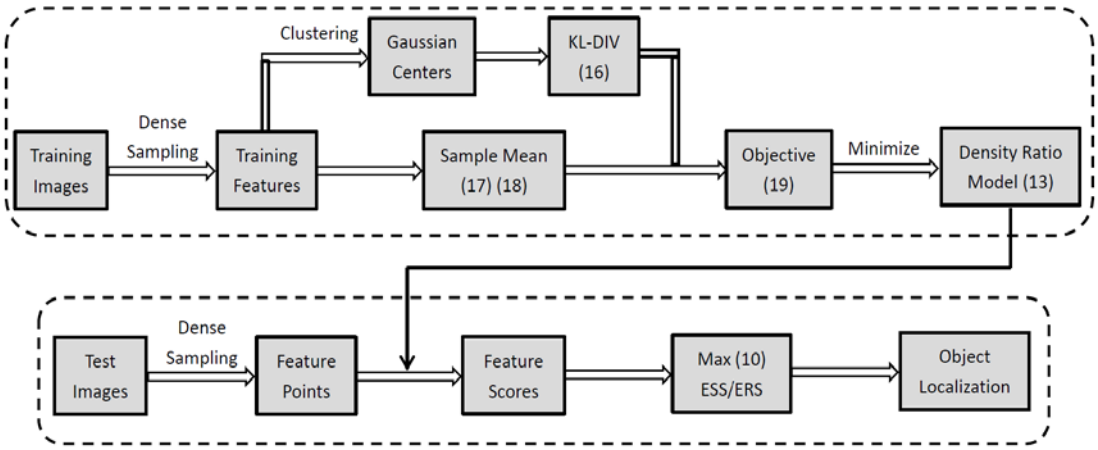


Fig. 2. The block diagram of the proposed architecture.

5. Experiments

In this section, we validate the performance of our method using a car dataset and PASCAL VOC 2007 dataset. Both the branch-and-bound and the branch-and-cut search strategies are employed to generate bounding-box and irregular-shape localizations. We compare with the baselines SVM-ESS [1], SVM-ERS [5], MIM-ESS [7], R-CNN [38], POS-FEAT [4], LS-MWCS [39] and GC_LP [32].

5.1 Component Analysis

In order to visualize the issues described in the introduction, and fully analyze the components in the proposed algorithm, we collected a small car dataset for experiments. It comprises 400 images of university campuses where half of them contain cars and the others do not. To protect privacy, all of the cars are presented in side viewpoint. The cars in the images vary in shape, size, grayscale intensity and location. The non-car images contain similar elements to the background of the car images. Some examples are given in **Fig. 3**. For each car image, we manually label the car by providing its minimum enclosing rectangle. In addition, following [40], we manually label the over-segmented regions to generate a pixel-wise ground-truth to evaluate irregular-shape localization performance.

Fig. 3. Some images from the campus car dataset.

We use half of the car images and non-car images as training data. The same as [2, 1, 5, 7], we only use the left images that actually contain a car for test. For each image, we extract SIFT [13] features densely using VL-FEAT toolbox [41]. For SVM-ESS [1], we cluster the training feature-points to get 1000 centers to construct bag-of-words representation. For our method, we use the same 1000 centers to center the Gaussian kernels.

We first examine the effect of the bias item in SVM-ESS and visualize the issue in localization by SVM classification. We train a linear SVM using the training data and apply it to test images for car localization. The performance is evaluated by average precision (AP) calculated from Precision-Recall (PR) curve. In the experiments, the bias item cannot be ignored, by contrast, it plays an important role in localization. When the bias item is totally ignored as suggested in the original work [1, 42], the AP is 0. None of the cars are correctly localized since the feature-weights are negatively biased seriously. When the feature-weights are tuned with a fraction of the bias value, the localization result can be improved. However, there is not a theoretical way to determine the coefficient. **Fig. 4** shows the changes of the AP when different coefficient $\frac{1}{n_b}$ is imposed on the bias value b. We also quantitatively compare it with our method. The PR curves are shown in **Fig. 5**. For SVM-ESS, we have tuned with the bias term to generate the largest AP. As can be seen from the figure, our method outperforms the baseline.

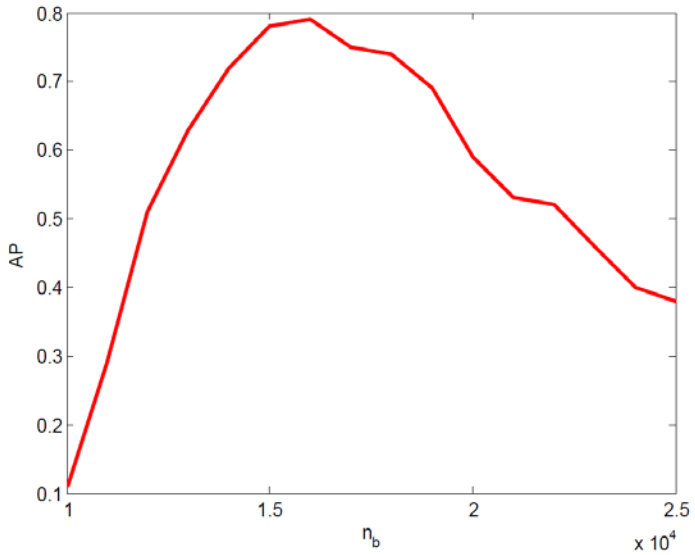


Fig. 4. AP changes along with different n_b.

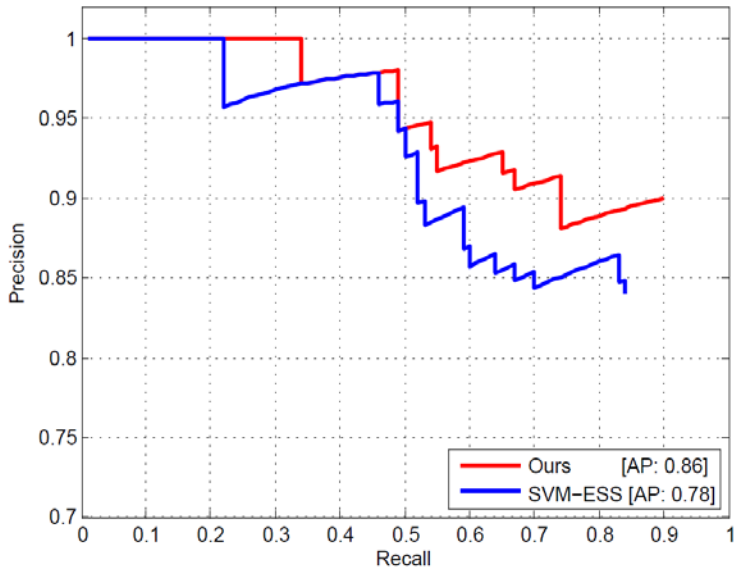


Fig. 5. Precision-recall curves on the Car dataset.

In order to analyze the failure of localization based on SVM classification and demonstrate the superiority of our feature-scoring strategy, we randomly choose some images from the dataset to visualize the feature scores and object localization in **Fig. 6**. We plot a red point to indicate a positive feature-score and green point negative feature-score. The brighter the color is, the larger the absolute value of the score is. The first column shows the original images. The second shows the feature scores from original SVM-ESS, and the third column corresponds to the SVM-ESS with the maximum AP whose weights are tuned with a fraction of the bias value. The last column shows our results. For each image, we also plot the localization bounding-box with blue rectangle to make further comparison. It can be seen that the original SVM weights are negatively biased: Most of the points are negatively scored even on the car, and the localization bounding-box shrinks to a tiny region. After we tune the weights with a fraction of

b, the weights are changed to the positive direction, and the localization results get much better. Since we model the density ratio using Gaussian kernels and estimate the feature score directly, the feature-point classification is more accurate and the localization results are visually better. One problem is that we use the bounding-box annotation for training, and much background clutter such as the curbs beneath the cars is included, consequently many feature-points in the curbs are misclassified during testing. However, it does not affect the localization, since the positive feature-points on the object dominate the bounding-box.

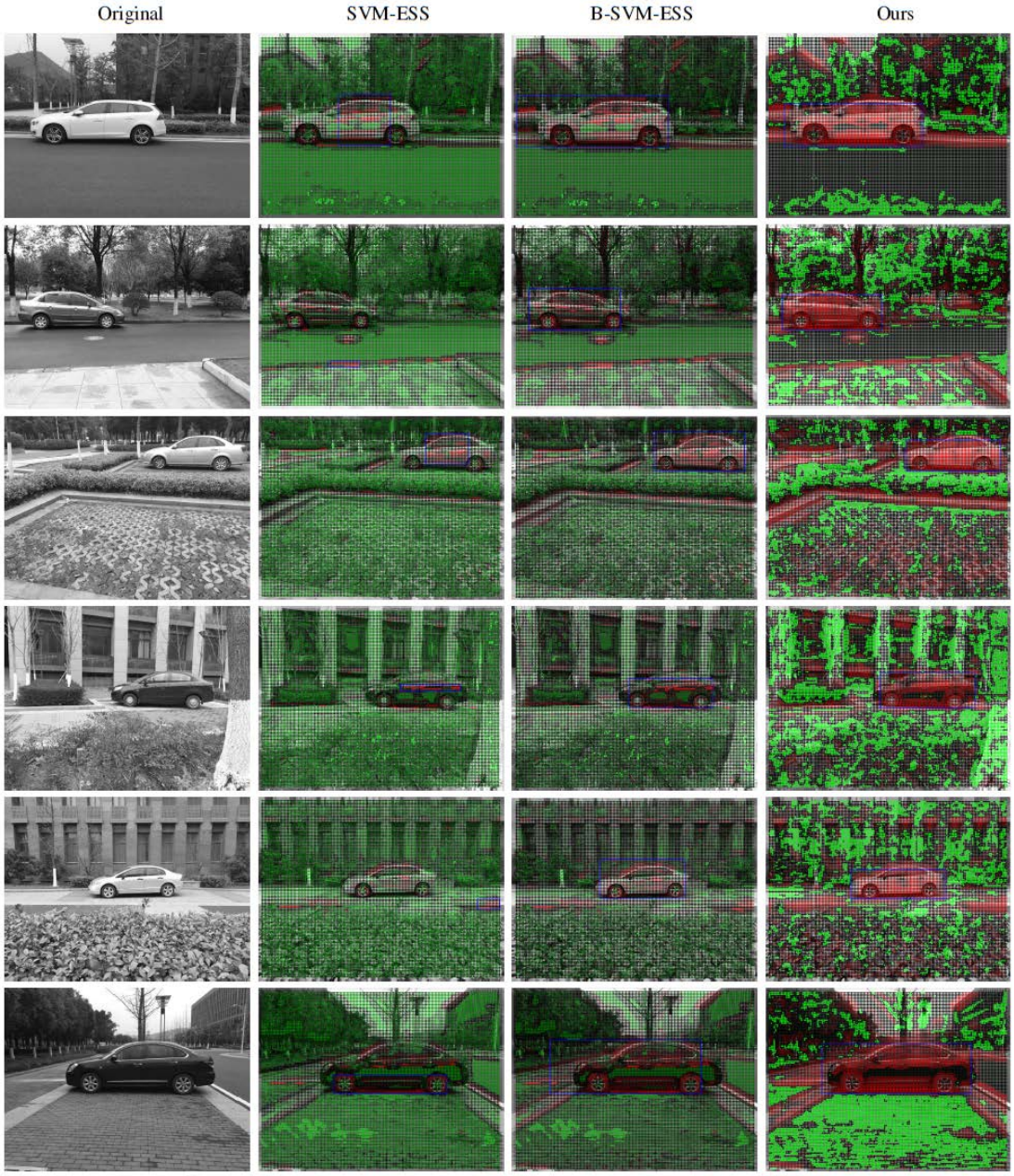


Fig. 6. Visualization of feature scoring and car localization. Each feature-point is stained with green and red colors for negative and positive weight values, respectively. The bounding-box localizations are presented with blue rectangles. The figure is best viewed in color.

Although we use the same branch-and-bound strategy for sub-window search, the time consumption varies for different feature scoring. When the feature scoring is more accurate, the searching is significantly fast. In our experiments, a desktop with Intel Core CPU (2.90GHz) and 4GB RAM with Windows 7 and Matlab R2014a was used as the experimental platform. The average searching time for SVM scoring is 4.8s per image (600*800 resolution), while that for our scoring is 0.3s.

5.2 Feature Voting for Object Localization

We evaluate the performance of our method for object localization in this section. We proposed an evaluation function and a feature-scoring strategy, and they are compatible with branch-and-bound as well as branch-and-cut searching, therefore we test both of them. PASCAL VOC 2007 is a more challenging and more popular dataset for object detection. The dataset has been previously partitioned into training and test sets. There are twenty classes of objects. All of the images are provided bounding-box annotation for localization evaluation. However, pixel-wise ground truth is not available. Vijayanarasimhan and Grauman [5] manually annotated the 'cat' and 'dog' classes to generate pixel-wise ground truth, which can be used to evaluate the localization results of branch-and-cut searching.

Table 1. Quantitative comparison on PASCAL VOC 2007. The performance is evaluated in terms of the mean average precision.

Methods	aero	bicy	bird	boat	bott	bus	car	cat	chai	cow
Ours	**0.340**	**0.270**	**0.277**	**0.267**	**0.109**	0.348	**0.378**	**0.466**	0.091	**0.269**
POS-FEAT	0.216	0.222	0.162	0.131	0.046	0.273	0.200	0.428	**0.103**	0.149
MIM-ESS	0.284	0.186	0.078	0.129	0.03	0.324	0.207	0.341	0.030	0.132
SVM-ESS	0.265	0.216	0.117	0.131	0.009	**0.388**	0.127	0.344	0.011	0.117

diningtable	dog	hors	moto	pers	pott	shee	sofa	trai	TV	ave
0.402	**0.353**	**0.391**	**0.302**	**0.203**	**0.145**	**0.152**	**0.347**	**0.445**	**0.109**	**0.283**
0.274	0.344	0.252	0.269	0.149	0.115	0.130	0.296	0.310	0.104	0.209
0.134	0.156	0.287	0.217	0.091	0.091	0.119	0.206	0.355	0.091	0.174
0.144	0.21	0.091	0.211	0.035	0.091	0.091	0.179	0.332	0.024	0.157

Table 2. Quantitative comparison for branch-and-cut search.

Categories	Ours	GC_LP	SVM-ERS	LS-MWCS
Car	0.679	0.615	0.583	0.607
PASCAL.cat	0.327	0.313	0.287	0.302
PASCAL.dog	0.316	0.292	0.269	0.285

For branch-and-bound searching, we consider SVM-ESS [1], MIM-ESS [7], POS-FEAT [4] as baselines. We train the models using training and validation images, and show the APs on the test set in **Table 1**, from which we can see our method performs better than the others on most classes. As we have found in the previous part, it is the DRE-based feature-scoring strategy that beats [1] and [7]. In addition, although the object searching aims to find a region with the densest positive score, negative feature-points still help to determine the extension of the region.

Our method is also compatible with branch-and-cut searching to generate irregular-shape object localization. We thus compare with related methods SVM-ERS [5], LS-MWCS [39] and GC_LP [32]. LS-MWCS is an extension of ERS to weak supervision. For fair comparison, all of the methods are trained using the same bounding-box. We list the mean overlap (mOL) in **Table 2**, and our method outperforms the baselines significantly. It demonstrates that our feature-scoring is better than that based on SVM classification for an object localization task.

We also visualize the irregular-shape localization results in **Fig. 7**. The first column shows the original images. The second column are the over-segmentation results for each image, where each segment is represented with different colors. The third and the last columns are the search results from SVM-ERS and ours, respectively. Our localization is visually better. Please note that the localization result is dependent on the over-segmentation quality. If the object parts are not separated from the background, the localization will be imprecise (e.g., the last row).

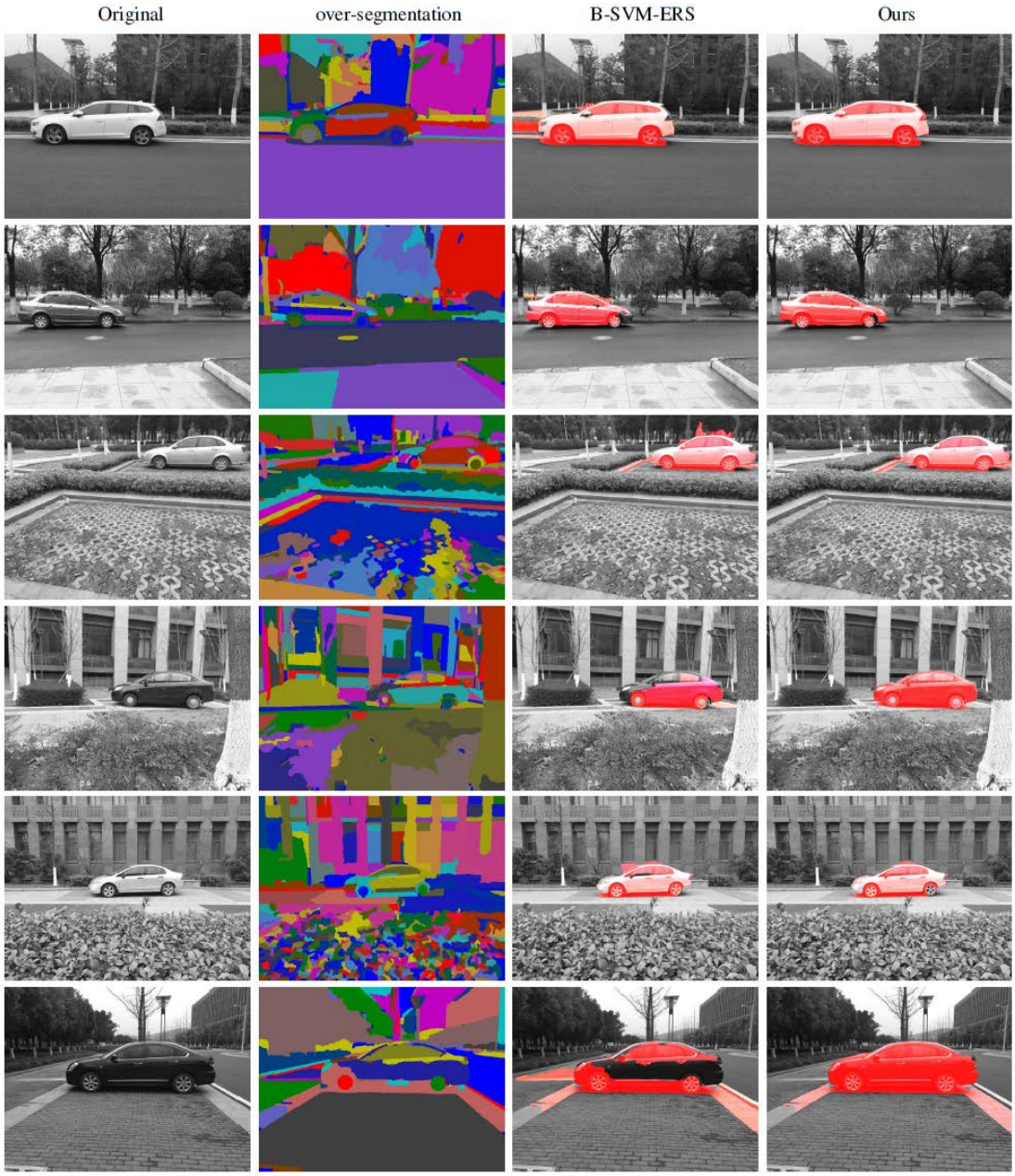


Fig. 6. Visualization of branch-and-cut search results. The figure is best viewed in color.

5.3 Deep Feature Voting

Classical local feature extractors (e.g., SIFT [13]) have recently been substantially outperformed by the introduction of the latest generation of CNNs [33]. In a typical CNN-based recognition algorithm, the output of the last layer is used as a feature representation for an image. This layer contains rich semantic information for classification, however, it is too coarse spatially to allow precise localization. By contrast, earlier layers contain more spatial information that is important for localization. To get the best of both worlds, some researchers have proposed concatenating outputs of multiple layers at a location to form a deep local feature representation [16, 17]. These previous works provide an opportunity for us to take advantage of deep local features in feature voting framework.

One popular way to incorporate DCNN in classical machine learning algorithms is to employ models pre-trained on large datasets such as ImageNet. We therefore consider the deep feature from the VGG-NET [43] pre-trained on the ImageNet. We first test on the car dataset. For each image, we first resize it to 224×224 and feed it into the pre-trained VGG model, then extract the feature maps of the convolutional layers in front of each pooling layer. Since the convolution and pooling operations in CNN reduce the spatial resolution, we up-sample each feature map to the scale of the original image. Each up-sampled feature map is considered as a feature representation for pixels from low level to high level. We do L_2 normalization for each feature map and concatenate them with coefficients to construct a 1472-dimensional hyper-column for each pixel. Based on the findings in [44], we empirically set the concatenation coefficients for each layer to 1, 0.8, 0.5, 0.3 and 0.1, respectively.

Table 3. Quantitative comparison of deep feature voting on PASCAL VOC 2007. The performance is evaluated in terms of the mean average precision.

Methods	aero	bicy	bird	boat	bott	bus	car	cat	chai	cow
fc_7	0.576	0.579	0.385	0.318	0.237	0.512	0.589	0.514	0.200	0.505
FT fc_7	0.642	0.697	0.500	0.419	0.320	0.626	0.710	0.607	0.327	0.585
FT fc_7 BB	**0.681**	**0.728**	0.568	0.430	**0.368**	0.663	**0.742**	0.676	**0.344**	0.635
Ours	0.672	0.710	**0.619**	**0.459**	0.293	**0.700**	0.709	**0.747**	0.334	**0.654**

diningtable	dog	hors	moto	pers	pott	shee	sofa	trai	TV	ave
0.409	0.460	0.516	0.559	0.433	0.233	0.481	0.353	0.510	0.574	0.447
0.465	0.561	0.606	0.668	0.542	0.315	0.528	0.489	0.579	0.647	0.542
0.545	0.612	0.691	**0.686**	0.587	**0.334**	**0.629**	0.511	0.625	**0.648**	0.585
0.606	**0.723**	**0.719**	0.657	**0.617**	0.228	0.581	**0.629**	**0.685**	0.585	**0.596**

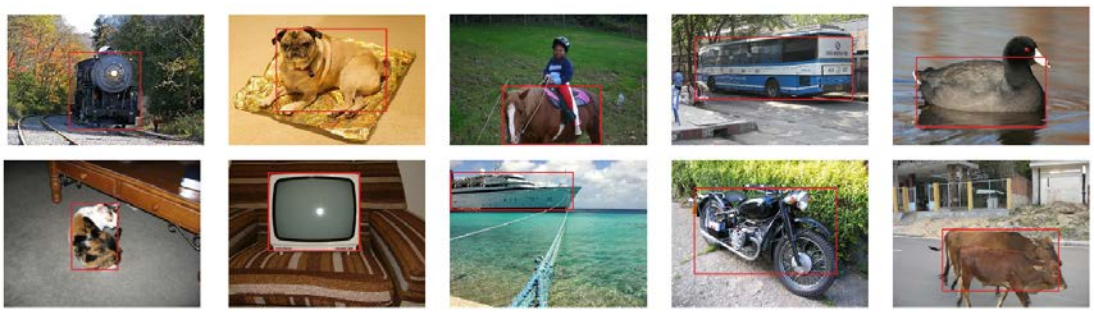


Fig. 8. Visualization of the deep feature voting on PASCAL VOC 2007.

When we replace the SIFT description by these deep features in feature voting, the localization AP achieves 1 on the car dataset, and every object are correctly localized. This demonstrates that by the aid of deep CNN description, feature-voting can do perfect object

localization. We also conduct experiments on PASCAL VOC 2007 to compare with deep CNN object detection methods fc_7, FT fc_7 and FT fc_7 BB in R-CNN [38]. We fine tune the pre-trained VGG-NET model on the training and validation images, then extract multiple layers as pixel descriptions. As shown in **Table 3**, the performance is significantly improved compared to the SIFT description, and is better than RCNN on average. Some localization results by the deep feature voting are visualized in **Fig. 8**.

6. Conclusion

Following the feature-voting framework, we formulate the object localization problem as a maximum posterior, and decompose the posterior probability of a region into feature-scores using Bayes' theorem and Naive-Bayes assumption. The feature-score is a function of density ratio. Instead of estimating the numerator and denominator probabilities separately, we adopt density ratio estimation techniques. Then efficient search strategies such as branch-and-bound and branch-and-cut can be used to locate objects in images. Since the feature-score implies intrinsic knowledge underlying category, our method outperforms the methods based on SVM classification and the methods based on nearest neighbour approximation. By the aid of deep CNN description, the localization performance of the feature-voting can be greatly improved.

There are three main advantages with respect to the proposed method. 1) By exploring the category knowledge of feature-points, our feature score is more accurate than that derived from SVM region classification, and thus leads to a better the localization performance. 2) Compared to the rectangular object localization methods, the feature-voting based method can generate localizations in the form of either bounding-box or pixel-mask. 3) The recently developed deep convolutional local features can be adopted in our method, and greatly improve the performance. However, the hand-crafted pipeline is not as efficient as the dominant end-to-end DCNNs.

References

[1] C. H. Lampert, M. B. Blaschko, and T. Hofmann, "Beyond sliding windows: Object localization by efficient subwindow search," in *Proc. of 2008 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2008. Article (CrossRef Link)

[2] C.-Y. Chen and K. Grauman, "Efficient activity detection with max-subgraph search," in *Proc. of 2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1274–1281, 2012. Article (CrossRef Link)

[3] A. Lehmann, B. Leibe, and L. van Gool, "Feature-centric efficient subwindow search," in *Proc. of IEEE International Conference on Computer Vision*, pp. 940–947, 2009. Article (CrossRef Link)

[4] H. Lei, G. Jiang, R. Wang, and L. Quan, "Object localization using positive features," *Neurocomputing*, vol. 171, pp. 463–470, 2016. Article (CrossRef Link)

[5] S. Vijayanarasimhan and K. Grauman, "Efficient region search for object detection," in *Proc. of 2011 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1401–1408, 2011. Article (CrossRef Link)

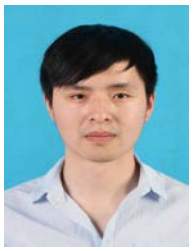
[6] T. Yeh, J. J. Lee, and T. Darrell, "Fast concurrent object localization and recognition," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 280–287, 2009. Article (CrossRef Link)

[7] J. Yuan, Z. Liu, and Y. Wu, "Discriminative subvolume search for efficient action detection," in *Proc. of 2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2442–2449, 2009. Article (CrossRef Link)

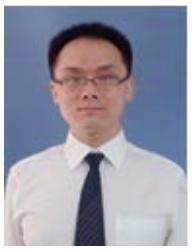
[8] C. Zhou and J. Yuan, "Arbitrary-shape object localization using adaptive image grids," in *Proc. of Computer Vision - ACCV 2012 - 11th Asian Conference on Computer Vision*, pp. 71–84, 2012. Article (CrossRef Link)
[9] O. Boiman, E. Shechtman, and M. Irani, "In defense of nearest-neighbor based image classification," in *Proc. of 2008 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2008. Article (CrossRef Link)
[10] R. Behmo, P. Marcombes, A. S. Dalalyan, and V. Prinet, "Towards optimal naive bayes nearest neighbor," in *Proc. of European Conference on Computer Vision*, pp. 171–184, 2010. Article (CrossRef Link)
[11] S. McCann and D. G. Lowe, "Local naive bayes nearest neighbor for image classification," in *Proc. of 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA*, pp. 3650–3656, June 16-21, 2012. Article (CrossRef Link)
[12] R. Timofte, T. Tuytelaars, and L. J. V. Gool, "Naive bayes image classification: Beyond nearest neighbors," in *Proc. of Computer Vision - ACCV 2012 - 11th Asian Conference on Computer Vision, Daejeon, Korea, Revised Selected Papers, Part I*, pp. 689–703, November 5-9, 2012. Article (CrossRef Link)
[13] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004. Article (CrossRef Link)
[14] H. Bay, T. Tuytelaars, and L. J. V. Gool, "SURF: speeded up robust features," in *Proc. of European Conference on Computer Vision*, pp. 404–417, 2006. Article (CrossRef Link)
[15] I. Laptev, "On space-time interest points," *International Journal of Computer Vision*, vol. 64, no. 2, pp. 107–123, 2005. Article (CrossRef Link)
[16] B. Hariharan, P. A. Arbeláez, R. B. Girshick, and J. Malik, "Hypercolumns for object segmentation and fine-grained localization," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA*, pp. 447–456, June 7-12, 2015. Article (CrossRef Link)
[17] C. Ma, J. Huang, X. Yang, and M. Yang, "Hierarchical convolutional features for visual tracking," in *Proc. of 2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chil*e, pp. 3074–3082, December 7-13, 2015. Article (CrossRef Link)
[18] M. Sugiyama, T. Suzuki, and T. Kanamori, *Density Ratio Estimation in Machine Learning*, Cambridge University Press, 2012. Article (CrossRef Link)
[19] A. D. Lehmann, B. Leibe, and L. J. V. Gool, "PRISM: principled implicit shape model," in *Proc. of British Machine Vision Conference*, pp. 1–11, 2009. Article (CrossRef Link)
[20] J. Yuan, Z. Liu, and Y. Wu, "Discriminative video pattern search for efficient action detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 9, pp. 1728–1743, 2011. Article (CrossRef Link)
[21] M. T. Dittrich, G. W. Klau, A. Rosenwald, T. Dandekar, and T. Müller, "Identifying functional modules in protein-protein interaction networks: an integrated exact approach," in *Proc. of International Conference on Intelligent Systems for Molecular Biology*, pp. 223–231, 2008.
[22] T. Kanamori, S. Hido, and M. Sugiyama, "Efficient direct density ratio estimation for non-stationarity adaptation and outlier detection," in *Proc. of Advances in Neural Information Processing Systems*, pp. 809–816, 2008.
[23] ——, "A least-squares approach to direct importance estimation," *Journal of Machine Learning Research*, vol. 10, pp. 1391–1445, 2009. Article (CrossRef Link)
[24] X. Nguyen, M. J. Wainwright, and M. I. Jordan, "Estimating divergence functionals and the likelihood ratio by convex risk minimization," *IEEE Transactions on Information Theory*, vol. 56, no. 11, pp. 5847-5861, 2010. Article (CrossRef Link)
[25] V. Vapnik, I. Braga, and R. Izmailov, "A constructive setting for the problem of density ratio estimation," in *Proc. of SIAM International Conference on Data Mining*, pp. 434–442, 2014. Article (CrossRef Link)
[26] M. Yamada, T. Suzuki, T. Kanamori, H. Hachiya, and M. Sugiyama, "Relative density-ratio estimation for robust distribution comparison," in *Proc. of Advances in Neural Information Processing Systems*, pp. 594–602, 2011.

[27] M. Sugiyama, M. Krauledat, and K. Müller, "Covariate shift adaptation by importance weighted cross validation," *Journal of Machine Learning Research*, vol. 8, pp. 985–1005, 2007.
[28] A. J. Smola, L. Song, and C. H. Teo, "Relative novelty detection," in *Proc. of International Conference on Artificial Intelligence and Statistics*, pp. 536–543, 2009.
[29] Y. Kawahara and M. Sugiyama, "Change-point detection in time-series data by direct density-ratio estimation," in *Proc. of SIAM International Conference on Data Mining*, pp. 389–400, 2009.
[30] S. Liu, M. Yamada, N. Collier, and M. Sugiyama, "Change-point detection in time-series data by relative density-ratio estimation," *Structural, Syntactic, and Statistical Pattern Recognition*, pp. 363-372, 2012. Article (CrossRef Link)
[31] C. Doersch, A. Gupta, and A. A. Efros, "Mid-level visual element discovery as discriminative mode seeking," in *Proc. of Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held, Lake Tahoe, Nevada, United States.*, pp. 494–502, December 5-8, 2013.
[32] L. Wang, J. Lu, X. Li, Z. Huan, J. Liang, and S. Chen, "Learning arbitrary-shape object detector from bounding-box annotation by searching region-graph," *Pattern Recognition Letters*, vol. 87, pp. 171–176, 2017. Article (CrossRef Link)
[33] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," in *Proc. of British Machine Vision Conference, BMVC 2014, Nottingham, UK*, September 1-5, 2014. Article (CrossRef Link)
[34] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 4, pp. 640-651, 2017. Article (CrossRef Link)
[35] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, 2017. Article (CrossRef Link)
[36] P. Zhang, D. Wang, H. Lu, H. Wang, and X. Ruan, "Amulet: Aggregating multi-level convolutional features for salient object detection," in *Proc. of IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy*, pp. 202–211, October 22-29, 2017. Article (CrossRef Link)
[37] M. Sugiyama, S. Nakajima, H. Kashima, P. von Bünau, and M. Kawanabe, "Direct importance estimation with model selection and its application to covariate shift adaptation," *Annals of the Institute of Statistical Mathematics*, vol. 60, no. 4, pp. 699-746, 2008. Article (CrossRef Link)
[38] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. of 2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA*, pp. 580–587, June 23-28, 2014. Article (CrossRef Link)
[39] J. Zhao, D. Meng, and J. Ma, "Density-based region search with arbitrary shape for object localization," *IET Computer Vision*, vol. 9, no. 6, pp. 943–949, 2015. Article (CrossRef Link)
[40] K. D. Tang, R. Sukthankar, J. Yagnik, and F.-F. Li, "Discriminative segment annotation in weakly labeled video," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2483–2490, 2013. Article (CrossRef Link)
[41] A. Vedaldi and B. Fulkerson, "VLFeat: An open and portable library of computer vision algorithms," in *Proc. of the 18th ACM international conference on Multimedia*, pp. 1469-1472, 2010. Article (CrossRef Link)
[42] C. H. Lampert, M. B. Blaschko, and T. Hofmann, "Efficient subwindow search: A branch and bound framework for object localization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 12, pp. 2129–2142, 2009. Article (CrossRef Link)
[43] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.
[44] X. Wei, J. Luo, J. Wu, and Z. Zhou, "Selective convolutional descriptor aggregation for fine-grained image retrieval," *IEEE Trans. Image Processing*, vol. 26, no. 6, pp. 2868–2881, 2017. Article (CrossRef Link)

Liantao Wang : received his B.S. degree in mechanical engineering, and Ph.D. degree in pattern recognition and intelligent system, both from Nanjing University of Science and Technology, in 2004 and 2015, respectively. From 2012 to 2014, he was a visiting scholar in the Robotics Institute at Carnegie Mellon University. He is currently an Assistant Professor with the College of Internet of Things Engineering at Hohai University. His research currently focuses on weakly supervised learning methods for object classification and localization.



Dong Deng : is currently an undergraduate student with the College of Internet of Things Engineering at Hohai University, Changzhou, China. His current research interests include image processing and object detection.



Chunlei Chen : received his bachelor degree, master degree and Ph.D. from Department of Automation at Northwestern Polytechnical University, Xian, China, in 2006, 2009 and 2014, respectively. He is currently a lecturer in School of Computer Engineering, Weifang University. His current research interests include machine learning and parallel computing.