Feature and Region Selection for Visual Learning

Ji Zhao*, Liantao Wang*, Ricardo Cabral, Fernando De la Torre

***Abstract*—Visual learning problems such as object classification and action recognition are typically approached using extensions of the popular bag-of-words (BoW) model. Despite its great success, it is unclear what visual features the BoW model is learning: Which regions in the image or video are used to discriminate among classes? Which are the most discriminative visual words? Answering these questions is fundamental for understanding existing BoW models and inspiring better models for visual recognition.**

To answer these questions, this paper presents a method for feature selection and region selection in the visual BoW model. This allows for an intermediate visualization of the features and regions that are important for visual learning. The main idea is to assign latent weights to the features or regions, and jointly optimize these latent variables with the parameters of a classifier (e.g., SVM). There are four main benefits of our approach: (1) Our approach accommodates non-linear additive kernels such as the popular χ^2 and intersection kernel; (2) our approach is able to handle both regions in images and spatio-temporal regions in videos in a unified way; (3) the feature selection problem is convex, and both problems can be solved using a scalable reduced gradient method; (4) we point out strong connections with multiple kernel learning and multiple instance learning approaches. Experimental results in the PASCAL VOC 2007, MSR Action Dataset II and YouTube illustrate the benefits of our approach.

***Index Terms*—Bag-of-words, feature selection, multiple kernel learning, multiple instance learning, weakly-supervised localization.**

I. Introduction

The last decade has witnessed great advances in machine learning and computer vision that have largely improved the performance and reduced the computational complexity of visual learning algorithms. Although there has been much progress in supervised visual learning, two main limitations still exist: (1) the reliance on human labeling limits the application of supervised methods in problems involving many categories; (2) these discriminative models lack interpretability because they do not produce mid-level representations (e.g., what are the most important visual features for discrimination?).

For instance, consider Fig. 1, where there are a set of images that contain a car (Fig. 1 (a)) and a set of images that do not contain a car (Fig. 1 (b)). Given these sets, the goal of a weakly-trained classifier is to discover discriminative regions and use them to train a car detector. Most of the successful approaches for weakly-supervised localization (WSL) [1], [2], [3], [4], [5] rely on bag-of-words (BoW). BoW approaches build a vocabulary of visual words to encode the visual representation and then use it to learn a binary classifier (e.g., kernel SVM). Although these techniques achieve state-of-the-art performance, the feature spaces induced by kernels obfuscate the understanding of which are the visual features that are most important for discrimination in the image space. The aim of this paper is to develop algorithms that learn in a weakly-supervised manner which are the discriminative features and regions. We aim to answer the following questions: Which visual words are used to discriminate cars versus non-cars (Fig. 1(c))? Which are the discriminative regions in the image (e.g., car in Fig. 1(d))? In addition to still images, we also apply our method to find discriminative spatio-temporal regions for activity recognition from video (Fig. 1 (e)-(h)).

WSL methods can partially solve the problem of localization of discriminative features, avoiding the time-consuming and error-prone manual localization process. Moreover, the selected regions are more informative to train detectors [1]. Due to its importance, WSL has been a popular topic researched in the last few years. Existing algorithms for WSL rely on multiple instance learning (MIL) and have mostly been applied to linear classifiers. A major challenge is how to extend these methods to cope with kernel representations while allowing for region and feature selection, which is a non-trivial task.

This paper proposes a feature and a region selection method for visual learning in the kernel space. The feature selection method is suitable for the family of additive kernels, and the region selection is valid for all kernels. The contributions of our work include: (1) a convex model for feature selection in the kernel space, and its application to find discriminative visual words; (2) a method for region selection using non-linear kernels, which can be used for the discovery and visualization of discriminative regions in images and spatio-temporal volumes in videos; (3) connections of our work with existing approaches including multiple kernel learning (MKL) and multiple instance learning (MIL). Experimental results in the PittCar dataset, PASCAL VOC 2007, MSR Action Dataset II and YouTube dataset illustrate the benefits of our approach.

II. Related Work

A. Feature Selection in Kernel Space

Selecting relevant features in kernel spaces has been a challenging problem addressed by several researchers. Cao et al. [6] developed a feature selection method by learning feature weights in the kernel space. This procedure is done as a data processing step, independently of the classifier construction. There also exist methods that perform feature selection and classifier construction jointly by inducing sparsity, such as [7], [8], [9]. Here the sparsity means sparse weight, which is usually realized by imposing L_1 norm constraints. We will

$*$ indicates equally contribution.

The authors are with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213, USA (e-mail: {zhaoji84, ltwang.nust}@gmail.com, rscabral@cmu.edu, ftorre@cs.cmu.edu).

L. Wang is also with the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China.

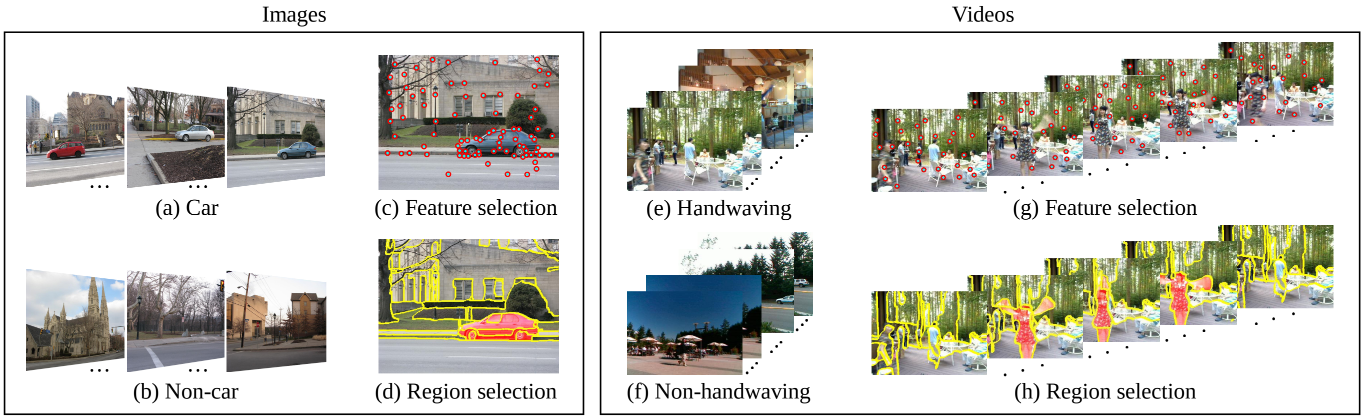


Fig. 1. Given a set of images containing a car (a) and images without a car (b), this paper proposes an algorithm to select the visual features (c) and regions (d) that are most discriminative in the kernel space. Similarly, given a set of videos containing hand-waving actions (e) and actions that are not hand-waving (f), we find the most discriminative spatio-temporal features (g) and spatio-temporal regions (h).

build on previous work by Nguyen et al. [10] who proposed a convex feature weighting method for linear SVM. Our work, however, extends [10] by adding non-linear additive kernels whose effectiveness have been validated in computer vision [11], [12].

B. Multiple Instance Learning (MIL)

In the MIL setting, each image is modeled as a bag of regions, and each region is an instance. With two classes, the negative bag only contains negative instances and the positive bag at least one positive. The goal of MIL is to label the positive instances within the positive bags. Many MIL algorithms have been successfully used for weakly-supervised learning, such as MILboost [13], MI-SVM [14], [1], [15], [4] and SparseMIL [16]. A convex MIL method named key-instance SVM (KI-SVM) is proposed in [17]. In addition to predicting bag labels, our approach can also locate regions of interest and it has been used in content-based image retrieval. MIL has been applied to object detection for images [15], [1], time series [1] and videos [2], [5], [18].

Among these methods, MI-SVM is arguably the most popular for WSL. However, current WSL methods based on MI-SVM have two main limitations: (1) most approaches use bounding boxes for localization (e.g., [1], [3]) instead of arbitrary shapes, and (2) most methods are limited to linear kernels. In this paper, our region selection method follows the idea of efficient region search (ERS) [19]: an object is a certain combination of several over-segmented regions, so it can localize objects with with arbitrary shape. Moreover, our region selection can take advantage of non-linear kernels.

C. Weakly Supervised Object Localization

Our region selection method aims to discover the discriminative regions in the positive images/videos, which turns out to be a way of weakly-supervised localization. In related work, Raptis et al. [20] used a latent SVM to classify videos using spatio-temporal patterns. Ghodrati et al. [21] improved action classification by refining the recognition and video segmentation iteratively in a coupled learning framework. CRANE [5] modified MIL by iterating through all of the negative segments, and each negative segment penalizes nearby segment in a positive video, improving existing algorithms. Weakly supervised localization also has a close relationship with the common pattern discovery from images that share common contents, such as co-segmentation and feature matching for sematic similar images [22], [23].

There are some works that enable bag-of-words to discover informative regions automatically, which are essential for visualization and image classification. For example, our work is most related to Liu and Wang [24], who proposed a region of support to visualize what the BoW model has learned. However, their method uses a linear SVM and it is unclear how to extend it to the kernel domain. Bilen et al. [25] proposed a semantic representation of an object and a new latent SVM to learn the spatial location of an object for enhanced image classification. However, this method is limited to linear kernel, and depends on a careful initialization. In addition, the localization is still limited to bounding-box, while our method yields arbitrary shape, the superiority of which has been stated in [19].

III. Feature Selection for Additive Kernels

This section proposes a convex feature selection method for additive kernels. Let $\mathcal{S} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ (see footnote[1] for an explanation of the notation used in this work) be a training set of n samples, where $\mathbf{x}_i \in \mathbb{R}^D$ is the histogram of BoW for the i^{th} image, D is the number of visual words in the codebook, and $y_i \in \{-1, +1\}$ are the corresponding labels.

Popular choices of kernels for visual learning are additive, such as the χ^2 and the histogram intersection kernels [12]. Formally, a kernel $K(\cdot,\cdot)$ on $\mathbb{R}^D \times \mathbb{R}^D$ is *additive* if it satisfies $K(\mathbf{x}_i, \mathbf{x}_j) = \sum_{k=1}^D \kappa(x_{ik}, x_{jk})$ for any samples $\mathbf{x}_i, \mathbf{x}_j \in \mathbb{R}^D$, where x_{ik} is the k^{th} bin of the BoW histogram for the i^{th} image. That is, the kernel function $\kappa(x_{ik}, x_{jk})$ is defined on one bin of the histogram.

[1] Bold lowercase letters, such as $\mathbf{p}$, denote column vectors. p_i represents the i^{th} entry of the column vector $\mathbf{p}$. Non-bold letters represent scalar variables. Calligraphic uppercase letters denote sets (e.g., $\mathcal{S}$, $\mathcal{B}$).

Given an additive kernel, the goal of feature selection is to weigh the features with a weight vector in the kernel space. We parameterize the feature space with a weight vector $\mathbf{p}$. That is, we construct a mapping $\boldsymbol{\phi}(\mathbf{x}_i, \mathbf{p}) = [\sqrt{p_1}\boldsymbol{\psi}^\top(x_{i1}), \cdots, \sqrt{p_D}\boldsymbol{\psi}^\top(x_{iD})]^\top$, that assigns different weights to different feature maps, where $\boldsymbol{\psi}(x_{ik})$ is the feature map for the k^{th} bin of the i^{th} histogram, $\mathbf{p} = [p_1, \cdots, p_D]^\top$ are the feature weights, and $p_k \geq 0\ \forall k$. In the maximum margin framework, we would like to find the separating hyperplane of a SVM and the feature weighting vector $\mathbf{p}$ that has the largest margin between classes. However, different values of $\mathbf{p}$ correspond to different feature spaces, and the margins in two different feature spaces cannot be directly compared, it is necessary to normalize the margin.

A. Normalized Margin SVM

Nguyen et al. [10] defined normalized margin as the ratio of the margin M over the square root of sum of squared distances (in the feature space) between same-class data instances. Formally, the *normalized margin* is defined as

$$\frac{M}{\sqrt{\sum_{i,j=1}^{n} \frac{1+y_i y_j}{2} \|\boldsymbol{\phi}(\mathbf{x}_i, \mathbf{p}) - \boldsymbol{\phi}(\mathbf{x}_j, \mathbf{p})\|^2}}. \tag{1}$$

Observe that the normalized margin is invariant to scale and translation in the feature space. The problem of finding the parameter $\mathbf{p}$ for the mapping and the parameters of the separating hyperplane that provides the largest normalized margin can be stated as

$$\begin{aligned} \max_{\overline{\mathbf{w}}, \bar{b}, M, \mathbf{p}} \quad & \frac{M}{\sqrt{\sum_{i,j=1}^{n} \frac{1+y_i y_j}{2} \|\boldsymbol{\phi}(\mathbf{x}_i, \mathbf{p}) - \boldsymbol{\phi}(\mathbf{x}_j, \mathbf{p})\|^2}} \\ \text{s.t.} \quad & y_i(\overline{\mathbf{w}}^\top \boldsymbol{\phi}(\mathbf{x}_i, \mathbf{p}) + \bar{b}) \geq M \quad \forall i; \\ & \|\overline{\mathbf{w}}\| = 1. \end{aligned} \tag{2}$$

We can see that if $\mathbf{p}$ is fixed, finding the hyperplane with the maximum normalized margin is equivalent to finding the hyperplane that maximizes the normal margin M.

Let $\mathbf{w} = \overline{\mathbf{w}}/M$, $b = \bar{b}/M$, and denote the normalization factor

$$\varphi(\mathbf{p}) = \sum_{i,j=1}^{n} \frac{1+y_i y_j}{2} \|\boldsymbol{\phi}(\mathbf{x}_i, \mathbf{p}) - \boldsymbol{\phi}(\mathbf{x}_j, \mathbf{p})\|^2. \tag{3}$$

Then $M = \|\overline{\mathbf{w}}\|/\|\mathbf{w}\| = 1/\|\mathbf{w}\|$. Substituting $\varphi(\mathbf{p})$ into problem (2), we obtain an equivalent problem

$$\begin{aligned} \max_{\mathbf{w}, b, \mathbf{p}} \quad & \frac{1}{\sqrt{\varphi(\mathbf{p})}\ \|\mathbf{w}\|} \\ \text{s.t.} \quad & y_i(\mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}_i, \mathbf{p}) + b) \geq 1 \quad \forall i. \end{aligned}$$

The above problem is again equivalent to

$$\begin{aligned} \min_{\mathbf{w}, b, \mathbf{p}} \quad & \frac{1}{2}\varphi(\mathbf{p})\|\mathbf{w}\|^2 \\ \text{s.t.} \quad & y_i(\mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}_i, \mathbf{p}) + b) \geq 1 \quad \forall i. \end{aligned}$$

Using soft-margin instead of hard-margin, the above formulation can be converted to

$$\begin{aligned} \min_{\mathbf{w}, b, \mathbf{p}, \boldsymbol{\xi}} \quad & \frac{1}{2}\varphi(\mathbf{p})\|\mathbf{w}\|^2 + C\sum_{i=1}^{n} \xi_i \\ \text{s.t.} \quad & y_i(\mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}_i, \mathbf{p}) + b) \geq 1 - \xi_i \quad \forall i; \quad \boldsymbol{\xi} \geq \mathbf{0}. \end{aligned} \tag{4}$$

Here, $\boldsymbol{\xi} = [\xi_1, \cdots, \xi_n]$ are slack variables which allow for penalized constraint violation, and C is the parameter that controls the trade-off between generalization and training error.

B. Normalized Margin SVM with Additive Kernels

In [10], they only solve the SVM with normalized margin for linear kernels. In this paper, we propose a method to solve the SVM with normalized margin for additive kernels in problem (4).

In order to transform problem (4) into a convex optimization problem and solve it efficiently, we make use of two properties of additive kernels. First, as we have mentioned, $\boldsymbol{\phi}(\mathbf{x}_i, \mathbf{p}) = [\sqrt{p_1}\boldsymbol{\psi}^\top(x_{i1}), \cdots, \sqrt{p_D}\boldsymbol{\psi}^\top(x_{iD})]^\top$ for additive kernel, so the normalization factor $\varphi(\mathbf{p})$ in Eq. (3) can be re-written as

$$\varphi(\mathbf{p}) = \sum_{k=1}^{D} a_k p_k, \tag{5}$$

where

$$\begin{aligned} a_k &= \sum_{i,j=1}^{n} \frac{1+y_i y_j}{2} \|\boldsymbol{\psi}(x_{ik}) - \boldsymbol{\psi}(x_{jk})\|^2 \\ &= \sum_{i,j=1}^{n} \frac{1+y_i y_j}{2} \left[\kappa(x_{ik}, x_{ik}) - 2\kappa(x_{ik}, x_{jk}) + \kappa(x_{jk}, x_{jk})\right] \end{aligned} \tag{6}$$

Note that a_k can be interpreted as the total distance of the k^{th} bin in kernel space, and it can be computed from the training data a priori. Other normalization factors can also be utilized without additional innovation. In [26], it provides a rather encyclopedic list of alternatives.

Second, the hyperplane $\mathbf{w}$ can be re-written as a vertical concatenation of D column vectors as $\mathbf{w} = [\mathbf{w}_1^\top, \cdots, \mathbf{w}_D^\top]^\top$, where each $\mathbf{w}_k$ weighs the feature map for each bin $\boldsymbol{\psi}(x_{ik})$. Then the following two equations hold: $\mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}_i, \mathbf{p}) = \sum_{k=1}^{D} \sqrt{p_k}\mathbf{w}_k^\top \boldsymbol{\psi}(x_{ik})$, and $\|\mathbf{w}\|^2 = \sum_{k=1}^{D} \|\mathbf{w}_k\|^2$.

Since $\varphi(\mathbf{p})$ is homogeneous in $\mathbf{p}$, we can always scale $\mathbf{p}$ appropriately to get $\varphi(\mathbf{p}) = 1$. Using this constraint, and making a variable substitution $\mathbf{w}_k \leftarrow \sqrt{p_k}\mathbf{w}_k$, problem (4) can be written as

$$\begin{aligned} \min_{\mathbf{w}, b, \mathbf{p}, \boldsymbol{\xi}} \quad & \frac{1}{2}\sum_{k=1}^{D} \frac{\|\mathbf{w}_k\|^2}{p_k} + C\sum_{i=1}^{n} \xi_i \\ \text{s.t.} \quad & y_i\left[\sum_{k=1}^{D} \mathbf{w}_k^\top \boldsymbol{\psi}(x_{ik}) + b\right] \geq 1 - \xi_i \ \forall i; \\ & \sum_{k=1}^{D} a_k p_k = 1; \quad \mathbf{p} \geq \mathbf{0}; \quad \boldsymbol{\xi} \geq \mathbf{0}. \end{aligned} \tag{7}$$

where we use the convention that $\frac{t}{0} = 0$ if $t = 0$ and ∞ otherwise. Problem (7) is convex, and we propose a scalable optimization strategy in Section III-D.

C. Relation to Multiple Kernel Learning

We note the remarkable relationship between our feature selection formulation in problem (7) and multiple kernel learning (MKL) [27], [28], [29], with the main difference being the constraints on $\mathbf{p}$. In MKL, the constraint is that $\mathbf{p}$ lies on the probability simplex, i.e., $\sum_{k=1}^{D} p_k = 1$ and $p_k > 0 \; \forall k$. In our feature selection formulation, the constraint is data-driven and adaptive, i.e., $\sum_{k=1}^{D} a_k p_k = 1$ and $p_k > 0 \; \forall k$. Weighing each bin differently will result in increased accuracy because normalized margin SVM is expected to assign higher weights to more informative bins. Besides, feature weighting can avoid the mis-domination of the bins with larger numeric ranges to those with smaller numeric ranges.

Our feature selection method used a normalized SVM margin for feature selection with additive kernels. By leveraging the properties of additive kernels, the normalized SVM margin is converted to a MKL alike problem. As a result, problem (7) can also be interpreted as a MKL with normalized margin to handle the feature scaling problem. There are some works that incorporate the radius of minimum enclosing ball (MEB) into MKL to address kernel scaling issue [30], [31]. Liu et al. [32] incorporated the radius information in a more robust and efficient way to avoid complex learning structure and high computational cost.

D. Optimization with the Reduced Gradient Method

The connection between our feature selection method and MKL allows us to exploit the existing algorithms for MKL. We can derive a scalable algorithm with proven convergence properties by optimizing problem (7) with a reduced gradient method [27]. For fixed $\mathbf{w}, b, \boldsymbol{\xi}$, problem (7) can be reformulated as a non-linear objective function with constraints over the simplex on $\mathbf{p}$. Formally,

$$\min_{\mathbf{p}} J(\mathbf{p}) \text{ such that } \sum_{k=1}^{D} a_k p_k = 1, \; p_k \geq 0, \tag{8}$$

where

$$J(\mathbf{p}) = \begin{cases} \min\limits_{\mathbf{w},b,\boldsymbol{\xi}} & \frac{1}{2}\sum_{k=1}^{D} \frac{\|\mathbf{w}_k\|^2}{p_k} + C\sum_{i=1}^{n} \xi_i \\ \text{s.t.} & y_i \left[\sum_{k=1}^{D} \mathbf{w}_k^\top \boldsymbol{\psi}(\mathbf{x}_{ik}) + b\right] \geq 1 - \xi_i; \forall i \\ & \boldsymbol{\xi} \geq \mathbf{0}. \end{cases} \tag{9}$$

To use a reduced gradient algorithm to optimize this problem, we first computed the gradient $\frac{\partial J}{\partial \mathbf{p}}$ and then calculate reduced gradient $\nabla_{\text{red}} J$ and descent direction $\mathbf{r}$ based on the gradient and constraints on $\mathbf{p}$.

To solve the problem, we introduced Lagrange multipliers α_i and β_i for the first and second constraints in problem (9), respectively. By setting the derivatives of the Lagrangian of problem (9) with respect to the primal variables $\mathbf{w}$, b, $\boldsymbol{\xi}$ to zero, we get the associated dual problem

$$\begin{aligned} \max_{\boldsymbol{\alpha}} \quad & -\frac{1}{2}\sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \sum_{k=1}^{D} p_k \kappa(x_{ik}, x_{jk}) + \sum_{i=1}^{n} \alpha_i \\ \text{s.t.} \quad & \sum_{i=1}^{n} \alpha_i y_i = 0; \quad 0 \leq \alpha_i \leq C \; \forall i. \end{aligned} \tag{10}$$

This dual problem is identified as the standard SVM dual problem using the combined kernel $K(\mathbf{x}_i, \mathbf{x}_j) = \sum_{k=1}^{D} p_k \kappa(x_{ik}, x_{jk})$. Because of strong duality, the objective value of this dual problem (10) is also $J(\mathbf{p})$. Existence and computation of derivatives of $J(\mathbf{p})$ have been discussed in previous literature [27]. Taking advantage of these previous works, the differentiation of the dual function with respect to p_k is

$$\frac{\partial J}{\partial p_k} = -\frac{1}{2}\sum_{i,j=1}^{n} \alpha_i^* \alpha_j^* y_i y_j \kappa(x_{ik}, x_{jk}) \; \forall k, \tag{11}$$

where α^* maximizes the objective function in problem (10).

Once the gradient of $J(\mathbf{p})$ is computed, $\mathbf{p}$ is updated using a descent direction ensuring that the equality constraint and the non-negativity constraints on $\mathbf{p}$ are satisfied. Let p_μ be the largest entry of $\mathbf{p}$. The reduced gradient of $J(\mathbf{p})$, denoted $\nabla_{\text{red}} J$, can be written as

$$[\nabla_{\text{red}} J]_k = \begin{cases} \frac{\partial J}{\partial p_k} - \frac{a_k}{a_\mu}\frac{\partial J}{\partial p_\mu} & \text{if } k \neq \mu; \\ \sum_{v \neq \mu} \left(\frac{a_v^2}{a_\mu^2}\frac{\partial J}{\partial p_\mu} - \frac{a_v}{a_\mu}\frac{\partial J}{\partial p_v}\right) & \text{if } k = \mu. \end{cases} \tag{12}$$

Descent direction is in the opposite direction with reduced gradient. However, the positivity constraints should be taken into account in the descent direction. If $p_k = 0$ and $[\nabla_{\text{red}} J]_k > 0$, using this descent direction would violate the positivity constraint for p_k. Thus, the descent direction for that component should be set to 0. Therefore, the descent direction for updating $\mathbf{p}$ is

$$\mathbf{r}_k = \begin{cases} 0 & \text{if } p_k = 0 \text{ and } \frac{\partial J}{\partial p_k} - \frac{a_k}{a_\mu}\frac{\partial J}{\partial p_\mu} > 0; \\ -\frac{\partial J}{\partial p_k} + \frac{a_k}{a_\mu}\frac{\partial J}{\partial p_\mu} & \text{if } p_k > 0 \text{ and } k \neq \mu; \\ \sum_{v \neq \mu, p_v > 0} \left(-\frac{a_v^2}{a_\mu^2}\frac{\partial J}{\partial p_\mu} + \frac{a_v}{a_\mu}\frac{\partial J}{\partial p_v}\right) & \text{if } k = \mu. \end{cases} \tag{13}$$

The usual updating scheme is $\mathbf{p} \leftarrow \mathbf{p} + \gamma \mathbf{r}$, where γ is the step size. γ is calculated using a line search method. For each γ during the line search, we obtained a new $\mathbf{p}$ and used an SVM solver to calculate problem (10).

We summarize the training of feature selection with additive kernels in Algorithm 1. For testing, the prediction function is

$$f(\mathbf{z}) = \sum_{i=1}^{n} y_i \alpha_i K(\mathbf{z}, \mathbf{x}_i) + b = \sum_{i=1}^{n} y_i \alpha_i \sum_{k=1}^{D} p_k \kappa(z_k, x_{ik}) + b.$$

Algorithm 1: Feature Selection: Normalized Margin SVM with Additive Kernels

```
Input: Training set $\mathcal{S} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$; kernel for each
       bin $\kappa(x_{ik}, x_{jk}) = \langle \boldsymbol{\psi}(x_{ik}), \boldsymbol{\psi}(x_{jk}) \rangle$; penalty
       coefficient $C$.
Output: Weight $\mathbf{p}$; SVM parameters $\boldsymbol{\alpha}$ and $b$.
Initialize $p_k = 1/\sum_{i=1}^{D} a_i, \forall k$;
Calculate $a_k$ by Eq. (6);
while stopping criterion not met do
    Solve problem (10) by an SVM solver to update $\boldsymbol{\alpha}$
    and $b$;
    Calculate $\frac{\partial J}{\partial \mathbf{p}}$ by Eq. (11);
    Set $\mu = \mathrm{argmax}_{\mu}\, \mathbf{p}_{\mu}$;
    Calculate the descent direction $\mathbf{r}$ by Eq. (12)(13);
    Line search along $\mathbf{r}$ to find the optimal step $\gamma$;
    Update $\mathbf{p} \leftarrow \mathbf{p} + \gamma \mathbf{r}$;
end
```

IV. Region Selection for Weakly Supervised Visual Learning

In the previous section, we have proposed a feature selection method in the kernel space for additive kernels. However, visual features are typically very sparse and it is difficult to assess which regions the classifier uses for learning. In this section, we propose a method for selecting discriminative regions in images and videos. Prior to applying our method, we over-segment the images and videos into regions, i.e. superpixels [33] or spatio-temporal regions [34]. Once the regions are segmented, we encoded each region using the BoW codebook learned from all training images/videos. We assumed an additive property of the classifier for region selection so that the classifier score of an image is a weighted sum of the score for each of the regions.

A. Weakly Supervised Localization as Region Selection

Given an over-segmentation for each image (or video) into m_i regions, $\mathbf{h}_{ik}$ and s_{ik} represent the BoW histogram and the importance (weight) for the k^{th} region in the i^{th} image. Our SVM for region selection minimizes

$$\begin{aligned} \min_{\mathbf{w}, b, \{\mathbf{s}_i\}, \boldsymbol{\xi}} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C_1 \sum_{i \in \mathcal{B}^+} \xi_i^+ + C_2 \sum_{i \in \mathcal{B}^-} \sum_{k=1}^{m_i} \xi_{ik}^- \qquad (14) \\ \text{s.t.} \quad & \sum_{k=1}^{m_i} s_{ik} \mathbf{w}^\top \boldsymbol{\phi}(\mathbf{h}_{ik}) + b \geq 1 - \xi_i^+ \;\; \forall i \in \mathcal{B}^+; \\ & -\mathbf{w}^\top \boldsymbol{\phi}(\mathbf{h}_{ik}) - b \geq 1 - \xi_{ik}^- \;\; \forall i \in \mathcal{B}^-, \\ & \qquad\qquad \forall k \in \{1, \cdots, m_i\}; \\ & \|\mathbf{s}_i\|_1 = 1, \mathbf{s}_i \geq \mathbf{0} \;\; \forall i \in \mathcal{B}^+; \;\; \boldsymbol{\xi} \geq \mathbf{0}. \end{aligned}$$

where $\boldsymbol{\phi}(\cdot)$ is the kernel feature map. $\mathcal{B}^+$ and $\mathcal{B}^-$ are index sets of training samples with label $+1$ and -1, respectively. C_1 and C_2 trade-off the model complexity and empirical losses on the positive and negative bags, respectively. The first constraint is imposed on the positive bags, and enforces that, for positive images, a combination of its segments' scores is expected to be positive or it will be penalized. The second constraint enforces that all the segments' scores of the negative images should be negative. The third constraint enforces that $\mathbf{s}_i$ lies on the probability simplex. Thus the solution tends to be sparse and can be used for region selection[2]. If we impose L_q norm constraint with $q > 1$ on $\mathbf{s}_i$, it will generate non-sparse solutions [29].

Prediction Once the SVM parameters are learned, the classification and localization for new test images can be performed simultaneously. Given the i^{th} image and its over-segmented regions (indexed by k), we can provide an initial estimate if a region belongs to a discriminative region or not by computing the decision value $\mathbf{w}^\top \boldsymbol{\phi}(\mathbf{h}_{ik}) + b$. The final score of the image is the weighed average score of its regions, that is, $\sum_k s_{ik} \mathbf{w}^\top \boldsymbol{\phi}(\mathbf{h}_{ik}) + b$. The weights s_{ik} are learned during training.

B. Relation to Multiple Instance Learning

The proposed region selection method has closed connection to multiple instance learning (MIL) algorithms. MIL makes the assumption that a negative bag contains only negative instances, whereas and a positive bag has at least one positive instance. However, in our region selection method, the bag label is determined by a combination of regions. This is a more reasonable assumption for visual learning because it is difficult to determine which region triggers a label for an image, considering that the segmentation may not yield perfect results. Generally speaking, in MIL, the label is determined by the maximum of the instances scores, while in our method, the label is determined by the weighted mean of all the instances' scores.

Our formulation is different from previous key-instance SVM (KI-SVM), where it is assumed that there is only one positive instance in each positive bag [17]. Our formulation is also different from kernel latent SVM (KLSVM) [4], which also relies on a single instance to determine the label for positive bags. In [35], the method scores an image using the combination of regions, but it is limited to the linear kernel case. Note that our region selection method in this section is compatible with any kernel.

C. Optimization with the Reduced Gradient Method

Similar to the feature selection problem (7), the region selection problem (14) can also be reformulated as a non-linear objective function with constraints over the simplex. We used the reduced gradient method to solve it with a coordinate descent strategy. First, we fixed the weights $\mathbf{s}$, and optimized the object function w.r.t. $\mathbf{w}$, b and $\boldsymbol{\xi}$. Second, we used the reduced gradient method to update $\mathbf{s}$.

In order to simplify the notation, we took each region in a negative image as a negative bag that contains only one

[2] In the paper, we refer to region as a set of superpixels in images or spatio-temporal regions in videos. The problem of (14) is one of region weighting. We call it region selection since the solution is sparse and only a few regions have non-zero weights.

instance. We set C_2 equal to C_1, and reformulate problem (14) as

$$\min_{\{\mathbf{s}_i\}} J(\{\mathbf{s}_i\}) \text{ such that } \|\mathbf{s}_i\|_1 = 1,\ \mathbf{s}_i \geq \mathbf{0}\ \forall i, \tag{15}$$

where

$$J(\{\mathbf{s}_i\}) = \begin{cases} \min\limits_{\mathbf{w},b,\boldsymbol{\xi}} & \dfrac{1}{2}\|\mathbf{w}\|^2 + C\sum\limits_{i=1}^{n}\xi_i \\ \text{s.t.} & y_i\left[\mathbf{w}^\top \sum\limits_{k=1}^{m_i} s_{ik}\phi(\mathbf{h}_{ik}) + b\right] \geq 1 - \xi_i\ \forall i \\ & \boldsymbol{\xi} \geq \mathbf{0}. \end{cases} \tag{16}$$

By setting the derivatives of the Lagrangian of problem (16) to zero, we get the associated dual problem

$$\begin{aligned} \max_{\boldsymbol{\alpha}} \quad & -\frac{1}{2}\sum_{i,j=1}^{n}\alpha_i\alpha_j y_i y_j \left(\sum_{k=1}^{m_i}\sum_{l=1}^{m_j} s_{ik}s_{jl}K(\mathbf{h}_{ik},\mathbf{h}_{jl})\right) \\ & + \sum_{i=1}^{n}\alpha_i \qquad (17) \\ \text{s.t.} \quad & \sum_{i=1}^{n}\alpha_i y_i = 0; \quad 0 \leq \alpha_i \leq C\ \forall i. \end{aligned}$$

This is the standard dual formulation for SVM with the combined kernel $K(\mathbf{h}_i, \mathbf{h}_j) = \sum_{k=1}^{m_i}\sum_{l=1}^{m_j} s_{ik}s_{jl}K(\mathbf{h}_{ik}, \mathbf{h}_{jl})$. Because of strong duality, $J(\{\mathbf{s}_i\})$ is also the objective value of this dual problem. By differentiating the dual function with respect to s_{ik}, we have

$$\frac{\partial J}{\partial s_{ik}} = -\frac{1}{2}\sum_{j=1}^{n}\alpha_i^*\alpha_j^* y_i y_j \sum_{l=1}^{m_j} s_{jl}K(\mathbf{h}_{ik},\mathbf{h}_{jl}), \tag{18}$$

where α^* maximizes problem (17). After the gradient $\frac{\partial J}{\partial s_{ik}}$ has been calculated, we can get the reduced gradient and descent direction using the way in Section III-D.

At first glance, computing the gradient in Eq. (18) seems to be computationally expensive. However, this calculation is efficient for the following reasons. First, we can reformulate it as a compact matrix formulation when calculating $\frac{\partial J}{\partial \mathbf{s}_i}$. Second, since α is sparse, the complexity of calculating gradient is largely reduced. The region selection method is sumarized in Table of Algorithm 2.

Algorithm 2: Region Selection Algorithm

```
Input: Training set $(\{(\mathbf{x}_{ik})\}_{k=1}^{m_i}, y_i)_i$, $i = 1, \cdots, n$;
       kernel $K(\cdot,\cdot)$; penalty coefficient $C$.
Output: Region annotation $\{(s_{ik})\}_{k=1}^{m_i}$, $i = 1, \cdots, n$;
        SVM parameters $\boldsymbol{\alpha}$ and $b$.
Initialize $s_{ik} = 1/m_i, \forall i, k$;
Construct block kernel matrix $\widetilde{\mathbf{K}}$. The $(k,l)$-th element of
    the $(i,j)$-th block is defined as $[\widetilde{\mathbf{K}}(i,j)]_{kl} = K(\mathbf{x}_{ik}, \mathbf{x}_{jl})$;
while stopping criterion not met do
    Calculate kernel matrix $\mathbf{K}$ with its element
        $\mathbf{K}_{ij} = \mathbf{s}_i^\top \widetilde{\mathbf{K}}(i,j)\mathbf{s}_j$;
    Calculate $J(\{\mathbf{s}_i\})$ in (16) by an SVM solver with
        kernel matrix $\mathbf{K}$, get SVM parameters $\boldsymbol{\alpha}$ and $b$;
    Calculate $\frac{\partial J}{\partial \mathbf{s}_i}$ for $i = 1, \cdots, n$ by Eq. (18);
    Calculate reduced gradient and descent direction
        $\mathbf{r}_i, \forall i$;
    Line search to find optimal step $\gamma_i$ for $\mathbf{s}_i$ $\forall i$;
    Update $\mathbf{s}_i \leftarrow \mathbf{s}_i + \gamma_i \mathbf{r}_i$, $i = 1, \cdots, n$;
end
```

V. Experimental Results

This section validated the performance of our feature selection and region selection algorithms by comparing them with other state-of-the-art approaches on the following four datasets:

PittCar Dataset [1] contains 400 images of which 200 are positive and 200 negative, see Fig. 2a. There is only one object in each positive image. Half of the positive and negative images were used as training data, and the rest were used for testing. For each image, we extracted SIFT features [36] densely and selected 10000 of them randomly. All the SIFT descriptors were quantized into 1000 visual words, obtained by applying K-means to 100000 training samples.

PASCAL VOC 2007 consists of 9963 images. For examples see Fig. 2b. There are 20 object categories, with some images containing multiple objects. This dataset has been previously split into training and testing sets, which contained 5011 and 4952 images respectively. We proceeded as in the PittCar Dataset, extracting SIFT features and building a codebook of 1000 dimensions.

MSR Action Dataset II [37] comprises 54 video sequences of crowded environments, see Fig. 2c. There are 3 action categories: hand waving, handclapping, and boxing. Each video sequence contains multiple actions. Following [18], we split each video to contain only one action and randomly selected 135 videos as training data and 46 for test data. During this random division, the videos containing multiple actions that could not be split temporally were always included in the testing set. We extracted STIP features [38] densely for each video. All the feature points were then quantized into 2000 words, which were obtained by applying K-means to 100000 training descriptors.

YouTube-Objects (YTO) [39] consists of videos collected from YouTube, see Fig. 2d. It contains 10 of the 20 classes in the PASCAL VOC. Tang *et al* [5] generated a ground truth set of 151 shots by manually annotating segments after the segmentation. We used the features in [5] that include histograms of dense-SIFT, histograms of RGB color, histograms of local binary patterns, histograms of dense optical flow, and heat maps.

A. Feature Selection Experiments

To validate the effectiveness of the proposed feature selection method, we compared our feature selection with χ^2 kernel with the following baselines: (i) Linear SVM; (ii) χ^2 kernel SVM; (iii) feature selection with linear SVM [10]; (iv) MKL using χ^2 kernel [27], due to their connection with our method explained in Section III. For MKL, each kernel is defined on one bin of the histograms.

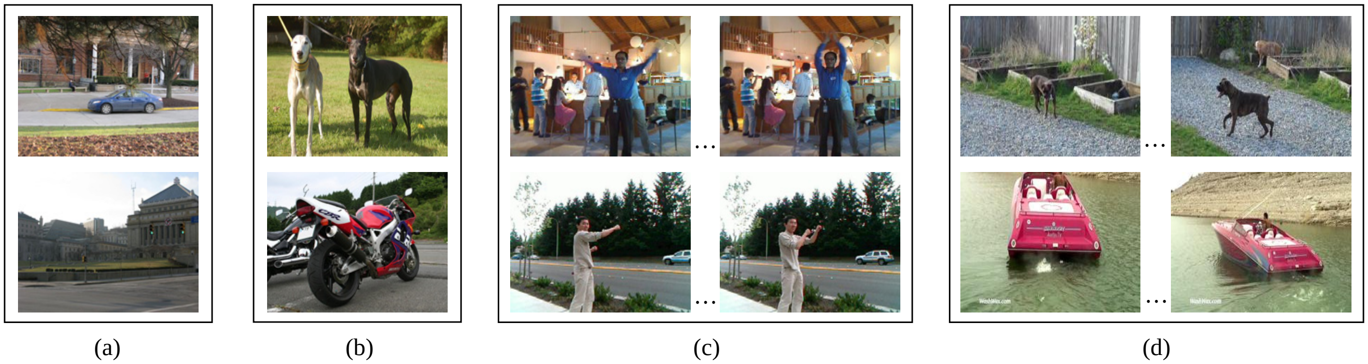


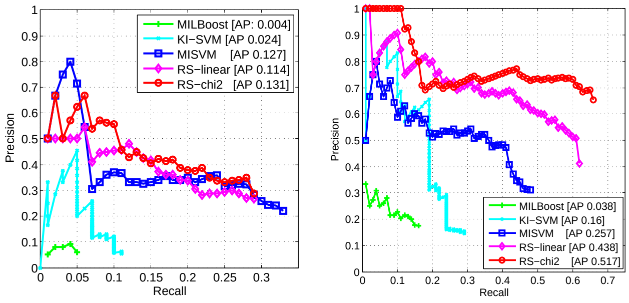
Fig. 2. Some examples of the datasets. (a) PittCar; (b) PASCAL VOC; (c) MSR Action II; (d) YouTube Objects.

For each method, parameters (e.g., C in the SVM) were chosen via cross-validation and we measured the classification performance using average precision (AP). To assess the complexity reduction achieved by feature selection, we also measured the number of selected features (i.e., non-zero weight). In this case, the features are the bins (clusters) in the BoW model. The results are presented in Table I for PittCar and MSR II datasets. We can see that our feature selection with χ^2 kernel (FS-χ^2) achieved the best average precision (AP) in all cases except 'Boxing', where it is outperformed only by the linear kernel, while the number of our selected features is significantly smaller than the original feature dimension. In Table II for PASCAL VOC 2007 dataset, the feature selection for χ^2 kernel SVM achieved comparable mean AP than χ^2 SVM (0.373 vs 0.375) over 20 classes, but used much less features (265 vs 1000).

A major goal of the paper is to illustrate that by performing feature and region selection, we can achieve a better interpretability of the BoW model. We visualized the selected visual words in the codebook for PittCar and PASCAL VOC 2007 datasets, in Fig. 3. From the feature selection results on the PittCar dataset, we can see that the most discriminative features mainly come from the wheels and doors of the cars. Note that the visual word with the fourth largest weight corresponds to the trunks of trees and fences. This is because trees occur more frequently in negative images than in positive images. As a result, this visual word is selected as a discriminative. For the cat and dog classes in PASCAL VOC dataset, several words latch on to cat and dog faces, while other visual words represent context (e.g., carpets) in which these animals usually appear. Since our method allows us to visualize the patches of visual words with their weights, the irrelevant words can be easily interpret by looking at the images in the dataset. From this example, we can see that feature selection can reveal which context the classifier is using for discriminating among classes.

B. Region Selection Experiments

As mentioned in Setion IV, region selection requires over-segmenting the images and videos first. For images, we used a hierarchical image segmentation to obtain superpixels [33]. For action localization on the MSR Action II, we followed [34] and used a regular voxel segmentation. For object localization on YTO dataset, we used the streaming hierarchical segmentation method of [40] to get supervoxels.



(a) overlap threshold is 0.5

(b) overlap threshold is 0.4

Fig. 5. Localization performance on the PittCar dataset.

PittCar: Due to the connection of region selection to MIL approaches, we compared our region selection using linear and χ^2 kernels with three popular MIL methods, MILboost [13], KI-SVM [17] and MI-SVM [1], on the PittCar dataset. We visualized the localization results in Fig. 4, from which we can see our region selection is visually best among these methods. In contrast, MILboost locates fewer regions, KI-SVM usually includes disperse background regions, and MI-SVM tends to include much background even though the size constraint has been imposed [1].

To provide a quantitative measure for the localization performance, we compared all methods using precision-recall curves, as shown in Fig. 5. We used the area of overlap (AO) measure to evaluate the correctness of localization. For this criterion, a threshold t should be defined for AO to imply a correct detection. Usually, t is set as 0.5 [41]. However, this is unfair for methods that localize arbitrary shape, because the ground truth is a bounding box and such methods provide a shape mask, which can yield more accurate localization. We thus also set t to 0.4. The PR curves of different t values are shown in Fig. 5. We can see that our method and MI-SVM perform comparably when $t = 0.5$. For $t = 0.4$, the region selection method performs significantly better than the baselines. Also, our region selection method using χ^2 kernel performs better than with a linear kernel, which reinforces the usefulness of kernels in visual learning.

MSR Action II: Since it is unclear how to apply the MI-SVM proposed in [1] to video, we used the state-of-the-art

TABLE I
THE COMPARISON OF CLASSIFICATION PERFORMANCE FOR FEATURE SELECTION METHODS AND MKL ON THE PITTCAR AND MSR ACTION II DATASETS.

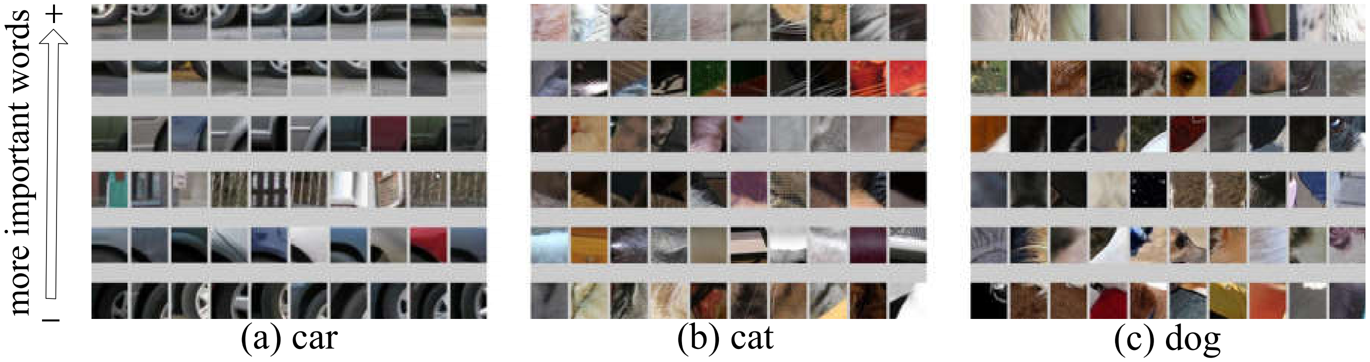
	PittCar		MSR Action II					
			Hand Clapping		Hand Waving		Boxing	
	AP	#Feat	AP	#Feat	AP	#Feat	AP	#Feat
linear SVM	0.833	1000	0.528	2000	0.630	2000	0.716	2000
χ^2 SVM	0.959	1000	0.563	2000	0.699	2000	0.680	2000
MKL-χ^2 [27]	0.961	120	0.687	102	0.741	96	0.810	112
FS-linear [10]	0.967	112	**0.717**	72	0.832	87	**0.897**	83
FS-χ^2 (ours)	**0.988**	56	**0.717**	79	**0.847**	56	0.852	45

TABLE II
THE COMPARISON OF CLASSIFICATION PERFORMANCE FOR FEATURE SELECTION METHODS AND MKL ON THE PITTCAR AND MSR ACTION II DATASETS.

	aeroplane		bicycle		bird		boat		bottle		bus		car	
	AP	#Feat	AP	#Feat	AP	#Feat	AP	#Feat	AP	#Feat	AP	#Feat	AP	#Feat
linear SVM	0.501	1000	0.274	1000	0.255	1000	0.418	1000	0.120	1000	0.249	1000	0.468	1000
χ^2 SVM	0.516	1000	0.384	1000	**0.295**	1000	**0.456**	1000	0.193	1000	**0.358**	1000	0.548	1000
MKL-χ^2 [27]	0.484	68	0.356	56	0.280	691	0.416	351	0.190	675	0.298	511	0.554	62
FS-linear [10]	0.392	364	0.314	397	0.241	661	0.323	358	0.147	396	0.239	350	0.535	422
FS-χ^2 (ours)	**0.517**	63	**0.397**	54	0.277	690	0.443	67	**0.198**	491	0.304	62	**0.565**	75

	cat		chair		cow		diningtable		dog		horse		motorbike	
	AP	#Feat	AP	#Feat	AP	#Feat	AP	#Feat	AP	#Feat	AP	#Feat	AP	#Feat
linear SVM	0.290	1000	0.343	1000	0.114	1000	0.245	1000	0.278	1000	0.427	1000	0.289	1000
χ^2 SVM	0.375	1000	0.338	1000	0.200	1000	**0.308**	1000	0.337	1000	**0.587**	1000	0.358	1000
MKL-χ^2 [27]	0.381	472	0.316	195	0.199	471	0.265	559	0.342	527	0.535	614	0.315	616
FS-linear [10]	0.315	665	0.355	384	0.186	443	0.228	665	0.306	769	0.431	379	0.295	376
FS-χ^2 (ours)	**0.384**	284	**0.366**	64	**0.215**	474	0.264	569	**0.347**	423	0.525	78	**0.378**	82

	person		pottedplant		sheep		sofa		train		TV monitor		AVERAGE	
	AP	#Feat	AP	#Feat	AP	#Feat	AP	#Feat	AP	#Feat	AP	#Feat	AP	#Feat
linear SVM	0.648	1000	0.122	1000	0.235	1000	0.225	1000	0.449	1000	0.252	1000	0.310	1000
χ^2 SVM	0.689	1000	**0.176**	1000	0.225	1000	**0.272**	1000	**0.566**	1000	0.330	1000	**0.376**	1000
MKL-χ^2 [27]	0.726	231	0.102	219	0.204	163	0.262	584	0.502	385	0.280	595	0.350	402
FS-linear [10]	0.697	484	0.113	420	0.226	282	0.243	596	0.420	525	0.294	372	0.315	465
FS-χ^2 (ours)	**0.741**	63	**0.176**	526	**0.236**	179	0.259	589	0.516	428	**0.341**	54	0.373	265



(a) car (b) cat (c) dog

Fig. 3. Patch visualization of top 6 visual words with highest weights in the feature selection. (a) Car in PittCar dataset, (b) Cat in PASCAL VOC 2007, (c) Dog in PASCAL VOC 2007. Each row line has 10 randomly selected patches corresponding to the visual word. From top to bottom, the weight changes from high to low.

method of Siva and Xiang [18] as a baseline.

As in the previous experiment, we used precision-recall curve to evaluate the localization performance quantitatively. To ensure comparability, we replicate the setup of [18] and set the temporal overlap to $1/8$ [34]. Qualitative and quantitative results are shown in Fig. 6 and Fig. 7 respectively. We can see that our region selection method using χ^2 kernel (RS-chi2) performs better than linear kernel (RS-linear). The region selection with a χ^2 kernel outperforms both MILboost and KI-SVM significantly and yields comparable results to Siva and Xiang [18]. Note, however, that our method is independent of the video-segmentation methods, whilst the method of Siva

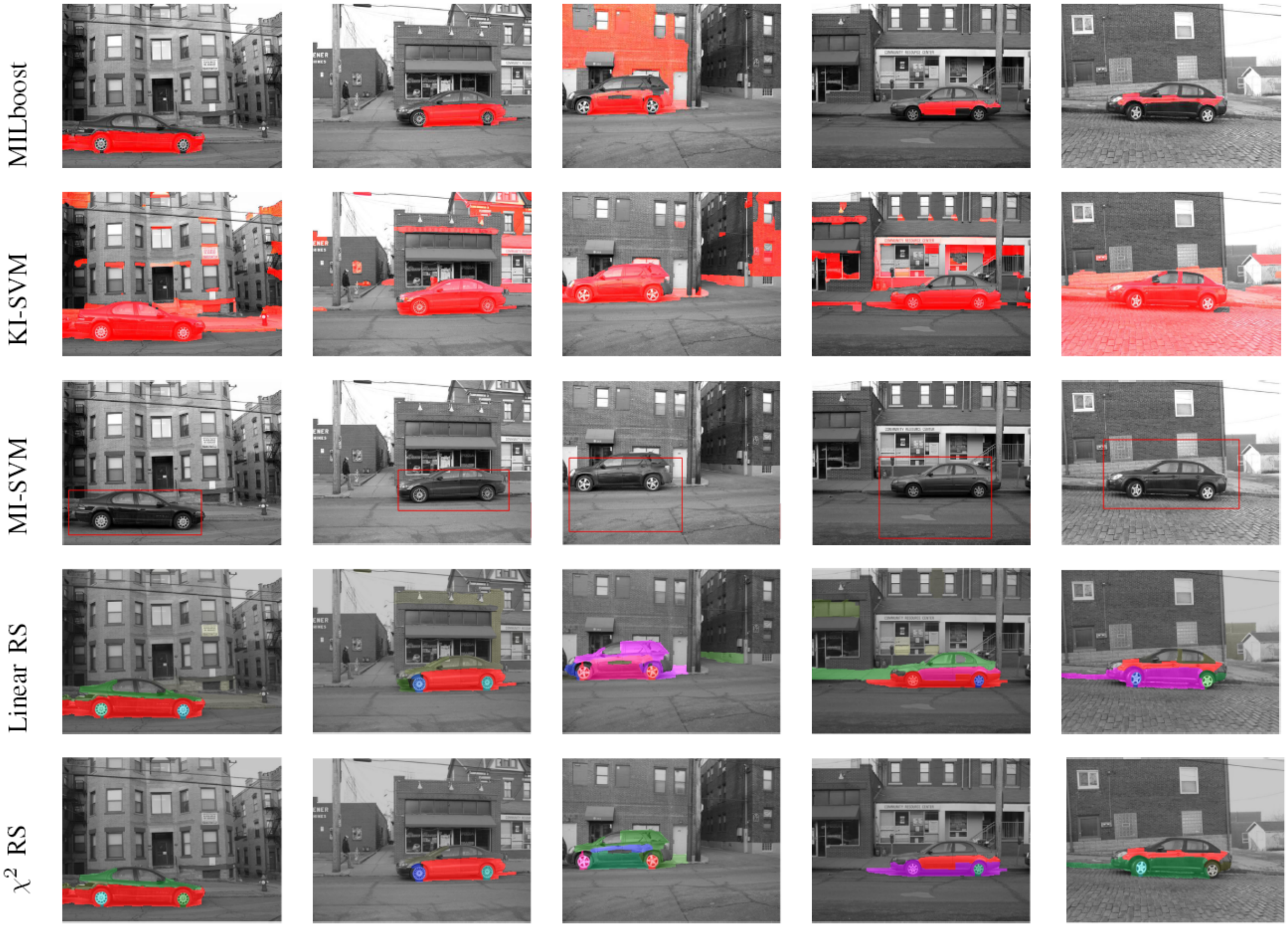


Fig. 4. Region selection for Pittsburgh Car dataset. For our method (rows three and four) the color encodes the weights of the selected regions (warmer means higher); only regions with positive weights are colored. Images best seen in color.

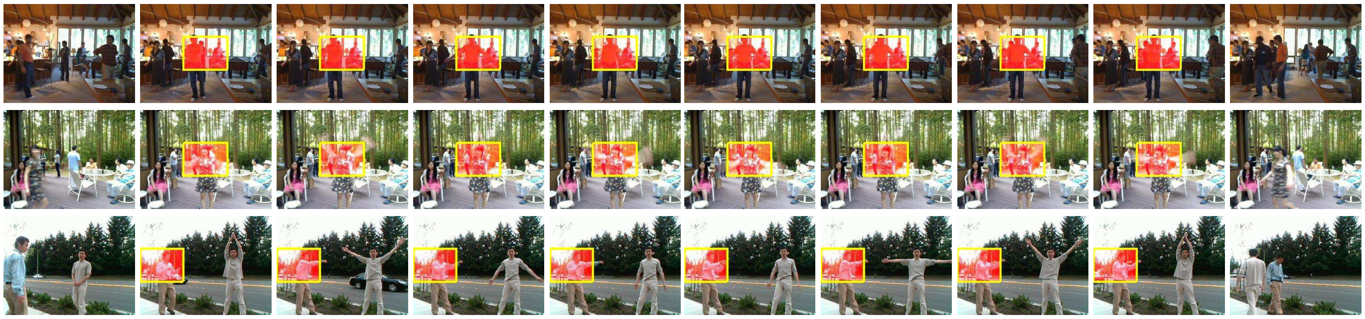


Fig. 6. Localization examples on MSR action II dataset. Each row corresponds to randomly selected 10 frames in a video. Yellow bounding boxes are the localized actions in the videos.

explicitly assumes the use of human detector.

YouTube-Objects: We also compared our region selection with CRANE [5] which is the state-of-the-art for object localization in videos. Here we use the χ^2 kernel in our method. The average precision for each class is shown in Tab. III. We can see that our method gets better results on most of the vehicle categories and gets worse results on animal categories. The reason may lie in the pre-segmentation. Since animals are often small in these videos and perform non-rigid motion, the segmentation method we used can not provide as good segmentation as that used in [5]. In general, however, our result is comparable to CRANE, which can be seen from the averaged PR curve over classes in Fig. 8. However, it is important to note that our method reported comparable results despite the fact that we used a worse segmentation algorithm.

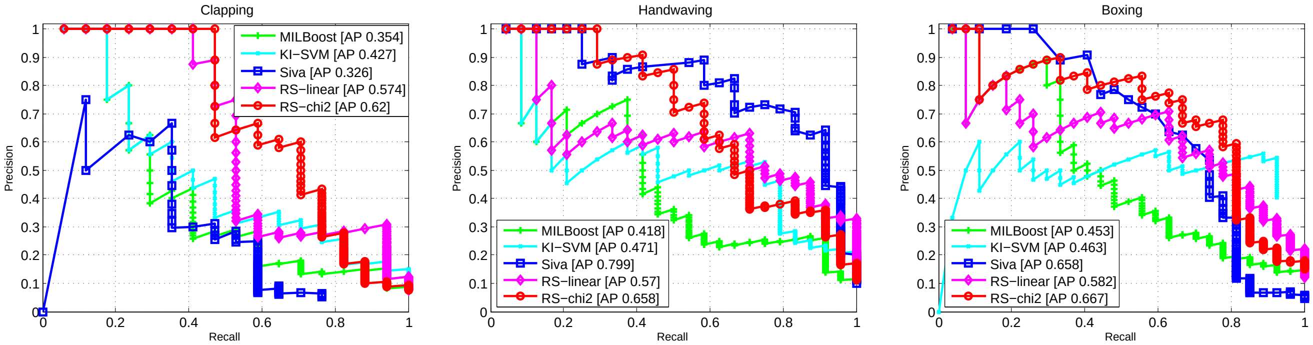


Fig. 7. Localization performance on MSR Action II.

TABLE III
AVERAGE PRECISION ON YOUTUBE-OBJECTS DATASET.

	aeroplane	bird	boat	car	cat	cow	dog	horse	motorbike	train	AVERAGE
CRANE [5]	0.365	**0.363**	**0.271**	0.446	**0.250**	**0.334**	**0.345**	**0.286**	0.158	0.204	**0.292**
Ours	**0.426**	0.279	0.268	**0.612**	0.204	0.203	0.283	0.148	**0.202**	**0.263**	0.289

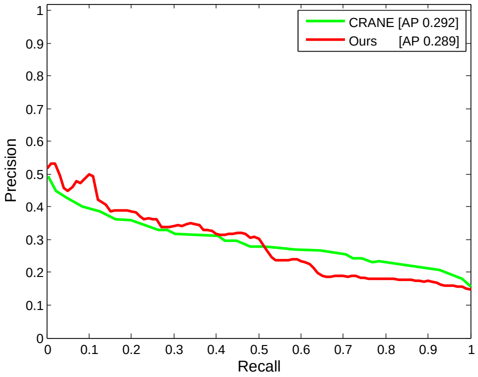


Fig. 8. Localization performance on YouTube-Objects dataset.

VI. CONCLUSIONS

This paper proposes a feature and region selection method for visualization and understanding of the bag-of-words model. These methods can also be used for image/video classification and weakly-supervised localization. A major advantage of our feature selection is that we can select features in the kernel space by solving a convex problem. This feature selection method achieves comparable accuracy to the state-of-the-art methods using significantly fewer number of features. In addition, our region selection method provides a tool to visualize the regions that the image/video classifier is weighting more aggressively to differentiate between class labels. The code is publicly available at https://sites.google.com/site/drjizhao.

While the method for feature selection is applicable to additive kernels, more research needs to be done to find convex solutions for non-additive kernels. In addition, other algorithms that can reduce the computational load of the optimization in space and time would be desirable. On the other hand, our region selection method can localize arbitrary shapes, beyond bounding-boxes; however, our method depends on the algorithm for over-segmentation and the object must be connected. These issues will remain to be explored in further research.

REFERENCES

[1] M. H. Nguyen, L. Torresani, F. De la Torre, and C. Rother, "Weakly supervised discriminative localization and classification: A joint learning process," in *Proc. Int. Conf. Comput. Vis.*, 2009, pp. 1925–1932.

[2] G. Hartmann, M. Grundmann, J. Hoffman, D. Tsai, V. Kwatra, O. Madani, S. Vijayanarasimhan, I. Essa, J. Rehg, and R. Sukthankar, "Weakly supervised learning of object segmentations from web-scale video," in *Proc. Eur. Conf. Comput. Vis. Workshop on Web-Scale Vision*, 2012, pp. 198–208.

[3] O. Russakovsky, Y. Lin, K. Yu, , and L. Fei-Fei, "Object-centric spatial pooling for image classification," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 1–15.

[4] W. Yang, Y. Wang, A. Vahdat, and G. Mori, "Kernel latent SVM for visual recognition," in *Advances in Neural Information Processing Systems 25*. Curran Associates, Inc., 2012, pp. 818–826.

[5] K. Tang, R. Sukthankar, J. Yagnik, and L. Fei-Fei, "Discriminative segment annotation in weakly labeled video," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 2483 – 2490.

[6] B. Cao, D. Shen, J.-T. Sun, Q. Yang, and Z. Chen, "Feature selection in a kernel space," in *Proc. Int. Conf. Mach. Learn.*, 2007, pp. 121–128.

[7] P. S. Bradley and O. L. Mangasaria, "Feature selection via concave minimization and support vector machines," in *Proc. Int. Conf. Mach. Learn.*, 1998, pp. 82–90.

[8] Y. Grandvalet and S. Canu, "Adaptive scaling for feature selection in SVMs," in *Advances in Neural Information Processing Systems 15*. MA, USA: MIT Press, 2002, pp. 553–560.

[9] J. Zhu, S. Rosset, T. Hastie, and R. Tibshirani, "1-norm support vector machines," in *Advances in Neural Information Processing Systems*. MA, USA: MIT Press, 2004, pp. 49–56.

[10] M. H. Nguyen and F. De la Torre, "Optimal feature selection for support vector machines," *Pattern Recognit.*, vol. 43, no. 3, pp. 584–591, 2010.

[11] K. Chatfield, V. Lempitsky, A. Vedaldi, and A. Zisserman, "The devil is in the details: an evaluation of recent feature encoding methods," in *Proc. British Machine Vision Conference*, 2011, pp. 76.1–76.12.

[12] A. Vedaldi and A. Zisserman, "Efficient additive kernels via explicit feature maps," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 3, pp. 480–492, 2012.

[13] P. Viola, J. C. Platt, and C. Zhang, "Multiple instance boosting for object detection," in *Advances in Neural Information Processing Systems 18*. MA, USA: MIT Press, 2005, pp. 1417–1424.

[14] S. Andrews, I. Tsochantaridis, and T. Hofmann, "Support vector machines for multiple-instance learning," in *Advances in Neural Information Processing Systems 15*. MA, USA: MIT Press, 2002, pp. 577–584.

[15] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part based models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, 2010.

[16] S. Vijayanarasimhan and K. Grauman, “Keywords to visual categories: Multiple-instance learning for weakly supervised object categorization,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2008, pp. 1–8.

[17] Y.-F. Li, J. T. Kwok, I. W. Tsang, and Z.-H. Zhou, “A convex method for locating regions of interest with multi-instance learning,” in *Proc. European Conference on Machine Learning*, 2009, pp. 15–30.

[18] P. Siva and T. Xiang, “Weakly supervised action detection,” in *Proc. British Machine Vision Conference*, 2011, pp. 1–11.

[19] S. Vijayanarasimhan and K. Grauman, “Efficient region search for object detection,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2011, pp. 1401–1408.

[20] M. Raptis, I. Kokkinos, and S. Soatto, “Discovering discriminative action parts from mid-level video representations,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 1242–1249.

[21] A. Ghodrati, M. Pedersoli, and T. Tuytelaars, “Coupling video segmentation and action recognition,” in *Proc. IEEE Winter Conf. Applications of Comput. Vis.*, 2014, pp. 618–625.

[22] A. Faktor and M. Irani, “Co-segmentation by composition,” in *Proc. Int. Conf. Comput. Vis.*, 2013, pp. 1297–1304.

[23] J. Ma, J. Zhao, and A. Y. Yuille, “Non-rigid point set registration by preserving global and local structures,” *IEEE Trans. Image Process.*, vol. 25, no. 1, pp. 53–64.

[24] L. Liu and L. Wang, “What has my classifier learned? Visualizing the classification rules of bag-of-feature model by support region detection,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 3586 – 3593.

[25] H. Bilen, M. Pedersoli, V. P. Namboodiri, T. Tuytelaars, and L. V. Gool, “Object classification with adaptable regions,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 3662–3669.

[26] Y. Feng and D. P. Palomar, “Normalization of linear support vector machines,” *IEEE Trans. Signal Processing*, vol. 63, no. 17, pp. 4673–4688, 2015.

[27] A. Rakotomamonjy, F. R. Bach, S. Canu, and Y. Grandvalet, “SimpleMKL,” *J. Mach. Learn. Res.*, vol. 9, pp. 2491–2521, 2008.

[28] P. V. Gehler and S. Nowozin, “Infinite kernel learning,” Max Planck Institute for Biological Cybernetics, Tech. Rep. TR-178, 2008.

[29] M. Kloft, U. Brefeld, S. Sonnenburg, and A. Zien, “ℓ_p-norm multiple kernel learning,” *J. Mach. Learn. Res.*, vol. 12, pp. 953–997, 2011.

[30] H. Do, A. Kalousis, A. Woznica, and M. Hilario, “Margin and radius based multiple kernel learning,” in *Proc. Eur. Conf. Mach. Learn.*, 2009, pp. 330–343.

[31] K. Gai, G. Chen, and C. Zhang, “Learning kernels with radiuses of minimum enclosing balls,” in *Advances in Neural Information Processing Systems 23*. Curran Associates, Inc., 2010, pp. 649–657.

[32] X. Liu, L. Wang, J. Yin, and L. Liu, “Incorporation of radius-info can be simple with simpleMKL,” *Neurocomputing*, vol. 89, pp. 30–38, 2012.

[33] P. Arbeláez, M. Maire, C. Fowlkes, and J. Malik, “Contour detection and hierarchical image segmentation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 898–916, 2011.

[34] C.-Y. Chen and K. Grauman, “Efficient activity detection with max-subgraph search,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 1274–1281.

[35] O. Yakhnenko, J. Verbeek, and C. Schmid, “Region-based image classification with a latent SVM model,” INRIA, Tech. Rep., 2011.

[36] D. Lowe, “Distinctive image features from scale-invariant keypoints,” *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

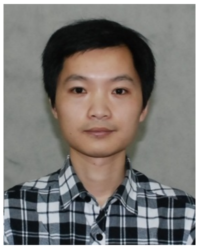
[37] J. Yuan, Z. Lin, and Y. Wu, “Discriminative video pattern search for efficient action detection,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 9, pp. 1728–1743, 2011.

[38] I. Laptev, “On space-time interest points,” *Int. J. Comput. Vis.*, vol. 64, no. 2, pp. 107–123, 2005.

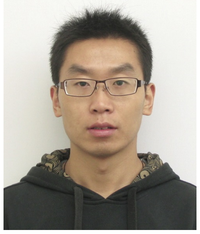
[39] A. Prest, C. Leistner, J. Civera, C. Schmid, and V. Ferrari, “Learning object class detectors from weakly annotated video,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 3282–3289.

[40] C. Xu, C. Xiong, and J. J. Corso, “Streaming hierarchical video segmentation,” in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 626–639.

[41] M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes (VOC) challenge,” *Int. J. Comput. Vis.*, vol. 88, pp. 303–338, 2009.



Ji Zhao received the B.S. degree in automation from the Nanjing University of Posts and Telecommunication and the Ph.D. degree in control science and engineering from Huazhong University of Science and Technology in 2005 and 2012, respectively. From 2012 to 2014, he was a Post-Doctoral Research Associate in the Robotics Institute, Carnegie Mellon University. His current research interests include image classification, image segmentation and kernel methods.



Liantao Wang received his B.S. degree in mechanical engineering from Nanjing University of Science and Technology, where he is currently pursuing the Ph.D. degree in Pattern recognition and intelligent system. From 2012 to 2014, he was a visiting student in the Robotics Institute at Carnegie Mellon University. His research currently focuses on weakly-supervised learning methods for object classification and localization.

Ricardo Cabral is a PhD student from Carnegie Mellon and IST-Lisbon. He received his Masters in Electrical and Computer Eng. at IST-Lisbon in 2009. He received an outstanding academic achievement award in 2008 from IST-Lisbon, where he worked in several projects, including video handling for the 2012 London Olympics. His research focuses on low rank models for computer vision and machine learning.

Fernando De la Torre is an Associate Research Professor in the Robotics Institute at Carnegie Mellon University. He received his B.Sc. degree in Telecommunications, as well as his M.Sc. and Ph. D degrees in Electronic Engineering from La Salle School of Engineering at Ramon Llull University, Barcelona, Spain in 1994, 1996, and 2002, respectively. His research interests are in the fields of computer vision and Machine Learning. Currently, he is directing the Component Analysis Laboratory (http://ca.cs.cmu.edu) and the Human Sensing Laboratory (http://humansensing.cs.cmu.edu) at Carnegie Mellon University. He has over 130 publications in referred journals and conferences. He has organized and co-organized several workshops and has given tutorials at international conferences on the use and extensions of Component Analysis.