Int J Comput Vis
DOI 10.1007/s11263-013-0646-8

Object and Action Classification with Latent Window Parameters

Hakan Bilen · Vinay P. Namboodiri · Luc J. Van Gool

Received: 1 October 2012 / Accepted: 18 July 2013

Abstract In this paper we propose a generic framework to incorporate unobserved auxiliary information for classifying objects and actions. This framework allows us to automatically select a bounding box and its quadrants from which best to extract features. These spatial subdivisions are learnt as latent variables. The paper is an extended version of our earlier work Bilen et al. (Proceedings of The British Machine Vision Conference, 2011), complemented with additional ideas, experiments and analysis. We approach the classification problem in a discriminative setting, as learning a max-margin classifier that infers the class label along with the latent variables. Through this paper we make the following contributions: (a) we provide a method for incorporating latent variables into object and action classification; (b) these variables determine the relative focus on foreground versus background information that is taken account of; (c) we design an objective function to more effectively learn in unbalanced data sets; (d) we learn a better classifier by iterative expansion of the latent parameter space. We demonstrate the performance of our approach through experimental evaluation on a number of standard object and action recognition data sets.

Keywords Object classification · Action classification · Latent SVM

This work was supported by the EU Project FP7 AXES ICT-269980.

H. Bilen (✉) · L. J. Van Gool
ESAT-PSI/iMinds, Ku Leuven, Kasteelpark Arenberg 10,
3001 Heverlee, Belgium
e-mail: hakan.bilen@esat.kuleuven.be

L. J. Van Gool
e-mail: luc.vangool@esat.kuleuven.be

V. P. Namboodiri
Alcatel-Lucent Bell Labs, Copernicuslaan 50, 2018
Antwerp, Belgium
e-mail: vinay.namboodiri@alcatel-lucent.com

L. J. Van Gool
Computer Vision Laboratory, ETH Zürich, Sternwartstrasse 7,
8092 Zurich, Switzerland

1 Introduction

In object detection, which includes the localization of object classes, people have trained their systems by giving bounding boxes around exemplars of a given class label. Here we show that the classification of object classes, i.e. the flagging of their presence without their localization, also benefits from the estimation of bounding boxes, even when these are not supplied as part of the training. The approach can also be interpreted as exploiting non-uniform pyramidal schemes. As a matter of fact, we demonstrate that similar schemes are also helpful for action class recognition.

In this paper we address the *classification* of objects (e.g. person or car) and actions (e.g. hugging or eating) (Pinz 2005) in the sense of PASCAL VOC (Everingham et al. 2007), i.e. indicating their presence but not their spatial/temporal localization (the latter is referred to as detection in VOC parlance). The more successful methods are based on a uniform pyramidal representation built on a visual word vocabulary (Boureau et al. 2010; Lazebnik et al. 2006; Wang et al. 2010). The focus then is often on the best features to use. In this paper, we augment the classification through an orthogonal idea, i.e. by adding more flexible spatial information. This will be formulated more generally as inferring additional unobserved or 'latent' dependent parameters. In particular, we focus on two such types of parameters:

Published online: 03 August 2013

– The first type specifies a cropping operation. This determines a bounding box in the image. This box serves to eliminate non-representative object parts and background.
– The second type specifies a splitting operation. It corresponds to a *non-uniform* image decomposition into 4 quadrants or temporal decomposition of a spatio-temporal volume into 2 video sub-sequences.

Apart from using these operations separately, we also study the effect of applying and jointly learning both these types of latent parameters, resulting in a bounding box which is also split. In any case, uniform grid subdivisions are replaced by more flexible operations.

At the time of our initial work (Bilen et al. 2011), there was earlier work using latent variables, but typically for object detection and not classification (Blaschko et al. 2010; Felzenszwalb et al. 2010; Vedaldi and Zisserman 2009). A notable exception is a contribution by Nguyen et al. (2009). They proposed a method for joint localization (only cropping) and classification. We believe that our learning approach is more principled however, and we go beyond cropping by also offering splits and crop + split combinations. This comes with improved results. Moreover, we propose iterative learning for these non-convex optimization problems, thereby more successfully avoiding local minima, as well as an objective function that can better deal with unbalanced data sets. In the meantime, the use of latent variables has gained traction in the area of classification(Bilen et al. 2012; Shapovalova et al. 2012; Sharma et al. 2012).

While it is possible to learn our latent variables by using a separate routine (Satkin and Hebert 2010), we adopt a principled max-margin method that jointly infers latent variables and class label. This we solve using a latent structural support vector machine (LSVM) (Yu and Joachims 2009). Self-paced learning has recently been proposed as a further extension for the improved learning of latent SVMs (Kumar et al. 2010), but was not used here. Instead, we explore an extension of the LSVM by initially limiting the latent variable parameter space and iteratively growing it. Moreover, we design a new objective function in the LSVM formulation to more effectively learn in the case of unbalanced data sets, e.g. when having a significantly higher number of negative images than positive ones. Those measures were observed to improve the classification results.

Our work can be seen as complementary to several alternative refinements to the bag-of-words principle. As a matter of fact, it could be combined with such work. For instance, improvements have also been obtained by considering multiple kernels of different features (Gehler and Nowozin 2009; Vedaldi et al. 2009). Another refinement has been based on varying the pyramidal representation step by considering maximal pooling over sparse continuous features (Boureau et al. 2010; Wang et al. 2010).

At a meta-level, recent progress in object classification has mainly been driven by the selection of more (sophisticated) features (Perronnin et al. 2010; Zhou et al. 2010). This has brought a couple of percentage points in terms of performance (Chatfield et al. 2011). Our improvements can actually be combined with those, and are shown here to bring similar improvements on their own. Yet, our approach does this at a lower computational cost.

As to action classification, this has mainly followed a bag of words approach as well. Early work towards classification of actions using space-time interest points (STIP) (Laptev and Lindeberg 2003) was proposed by Schüldt et al. (2004). A detailed evaluation of various features has been carried out lately by Wang et al. (2009).

In summary, the main contributions of this paper are (a) the introduction of latent variables for enhanced classification, (b) a principled technique for estimating them in the case of object and action classification, (c) adapted optimization to improve learning in the case of imbalanced data sets, and (d) the avoidance of local optima through an iteratively widened parameter space.

The remainder of the paper is structured as follows. Section 2 describes the latent parameter operations and how they are included in the overall classification framework. Section 3 explains the inference and learning procedures. Section 4 shows how the LSVM framework is adapted for imbalanced data sets. Section 5 introduces an iterative learning approach for these latent variables. Section 6 describes the results on standard object and action classification benchmarks and analyzes the statistical significance of the improved results. Section 7 concludes the paper.

2 Latent Operations

We explore how far information resulting from cropped or splitted regions can serve classification. In order to see what is meant by those crop and split operations, one can turn to Figs. 1 and 2 for the cases of single images (object classification) and videos (action classification), resp. Representative classification examples from the Graz-02 data set are shown in Figs. 3, 4, 5 and 6. We now discuss the two basic operations represented by our latent variables, cropping and splitting, in turn.

2.1 Crop

Our first latent operation builds on the motivation that including class related content and discarding irrelevant and confusing content should provide a better discriminant function for classification. For the sake of simplicity, we use a rec-

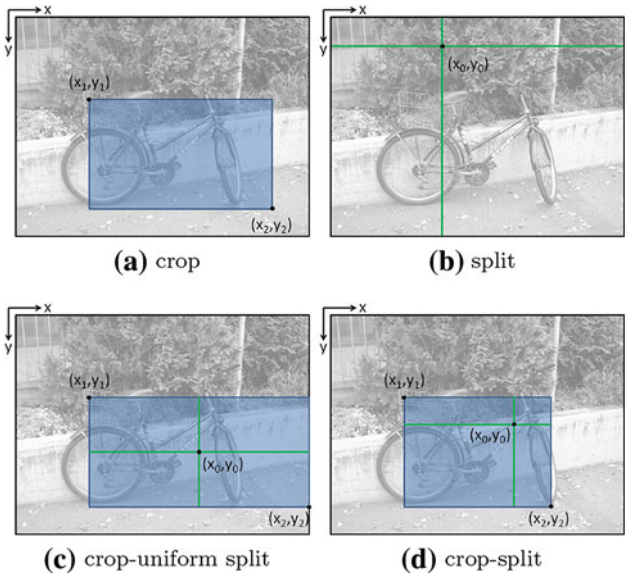


Fig. 1 Illustrative figure for latent operations, crop, split, crop-uniform split and crop–split on images. The crop–split operations have the most degree of freedom with six coordinates

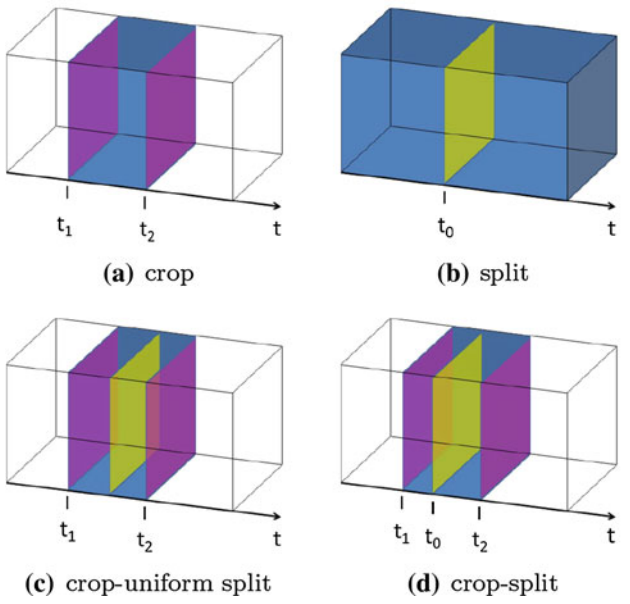


Fig. 2 Illustrative figure for latent operations, crop, split, crop-uniform split and crop–split on videos. Differently from spatial operations in images, the latent operations are performed only in temporal domain.

tangular bounding box to separate the two parts. The bounding box is represented by two points for both spatial and temporal cropping. We denote the latent parameter set with $h_{\text{crop}} = \{x_1, y_1, x_2, y_2\}$ and $h_{\text{crop}} = \{t_1, t_2\}$ for images and video sequences resp. Illustrations for cropping were shown in Figs. 1a and 2a.

For the Graz-02 3-class person-car-bike examples in Fig. 3, we illustrate the derived cropping operations with

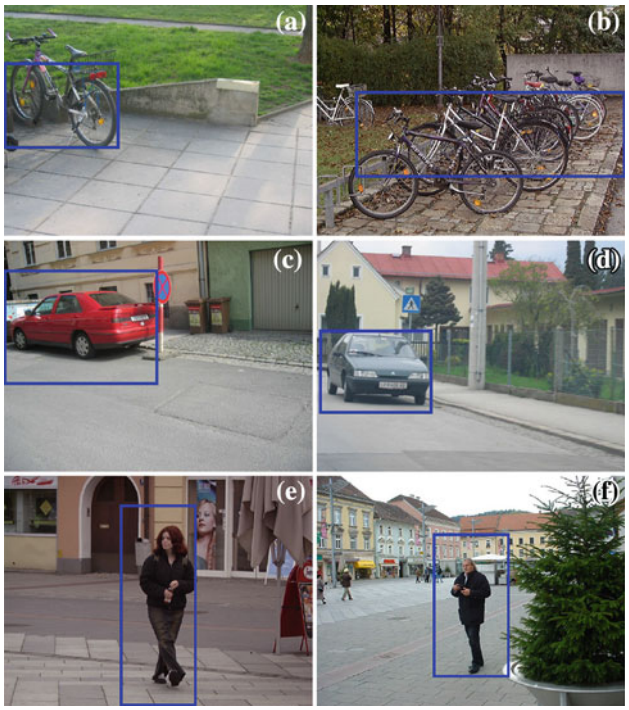


Fig. 3 Crop examples for different object categories from the Graz-02 data set : (**a**) shows the eliminated non-representative object parts, (**b**) shows cropped region in the presence of same class multiple objects, (**c**–**f**) depict included background context in the bounding boxes. While the 'road' contains the context information for 'car', it is 'road' and 'building' for the 'person'

blue drawn bounding boxes. Differently from object detection methods, our classification method is not required to localize objects accurately. Instead it can exploit bounding boxes to discard object parts that are not helpful in its particular classification task, while keeping the helpful ones in. The latter can very well include parts of the background (e.g. road for the car in Fig. 3c–d, building for the person in Fig. 3e, f). On the other hand, parts with too much variation in their appearance or with a high uncertainty of being picked up by the selected features, can be left out of the box. Also a bounding box is allowed to include more than one object of the same class (Fig. 3b).

2.2 Split

It is known that using pyramidal subdivisions of images or videos improves the classification of objects and actions (Laptev et al. 2008; Lazebnik et al. 2006). Therefore, it stands to reason to also consider a pyramid-type subdivision, but with added flexibility. Rather than splitting an image uniformly into equal quadrants, we consider splitting operations that divide into unequal quadrants. In the same vein, we allow a video fragment to be temporally split into two subsequences, which are not halves. In contradistinction with

cropping where all further analysis is confined to the selected bounding box, we will use all splitted portions as well as the entire image or video, i.e. a total of five portions for images and three for videos.

Note that in this paper we only consider a single layer of subdivision of the pyramid, the extension to multi-layer pyramids is not covered yet. Hence, our splits are fully characterized by one point. We denote the latent variable set with $h_{\text{split}} = \{x_0, y_0\}$ (Fig. 1b) and $h_{\text{split}} = \{t_0\}$ (Fig. 2b) for images and videos, resp.

We show splitting samples for the bike, car and person classes with green crossing lines in Fig. 4. We observe that bikes are often located in the left and right bottom cells, while cars and people are usually splitted into four 'quadrants'.

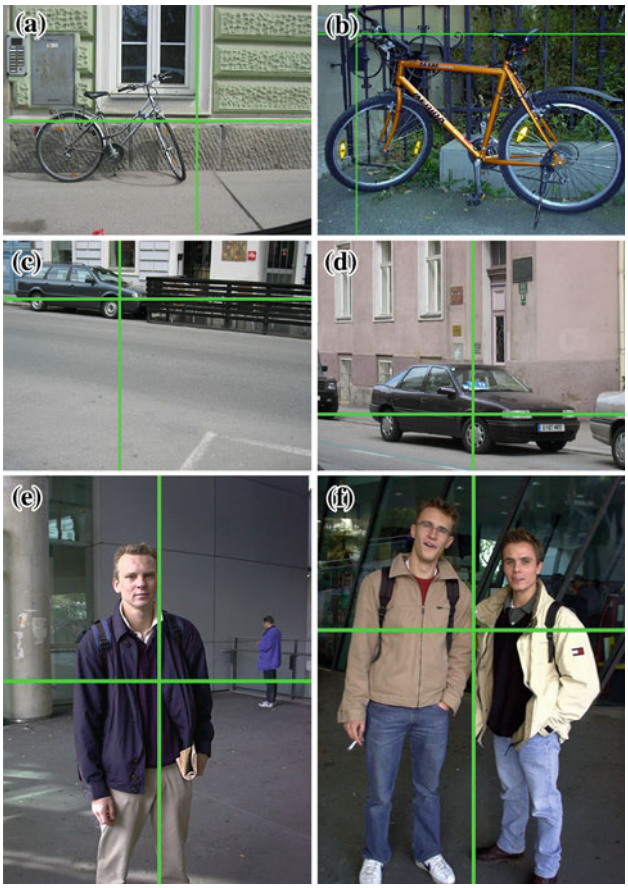


Fig. 4 Representative split examples for the bike, car and person classes from the Graz-02 data set. The wheels of bikes in the shown images (**a**) and (**b**) are contained in the *bottomleft* or *right* subdivisions. Splitting aligns the whole scene between (**c**) and (**d**) examples. The *upper* quadrants contain buildings and windows of cars, while the *lower* ones contain road and wheels of cars. Since the split operation can only split whole image into four divisions, it cannot exclude non-representative parts of images. In case of multiple objects, splitting point can move to the visually dominant one (person) as in (**e**) or to between two similar size objects (people) as in (**f**).

2.3 Crop - Uniform Split

Our crop-uniform split operation learns a cropped region, which is then subdivided further into equal parts, in order to enrich the representation in pyramid-style. The latent parameter set is that of cropping. The combined operation is illustrated in Figs. 1c and Fig. 2c. We illustrate crop-uniform splitting examples with blue cropping boxes and green uniform splits in Fig. 5. Figure 5 heralds more effective model learning than through uniform splitting only. The richer representation of cropping and uniform splitting will in section 6 be seen to outperform pure cropping.

2.4 Crop–Split

The combined crop–split operation comes with the highest-dimensional latent parameter set of all four cases stud-

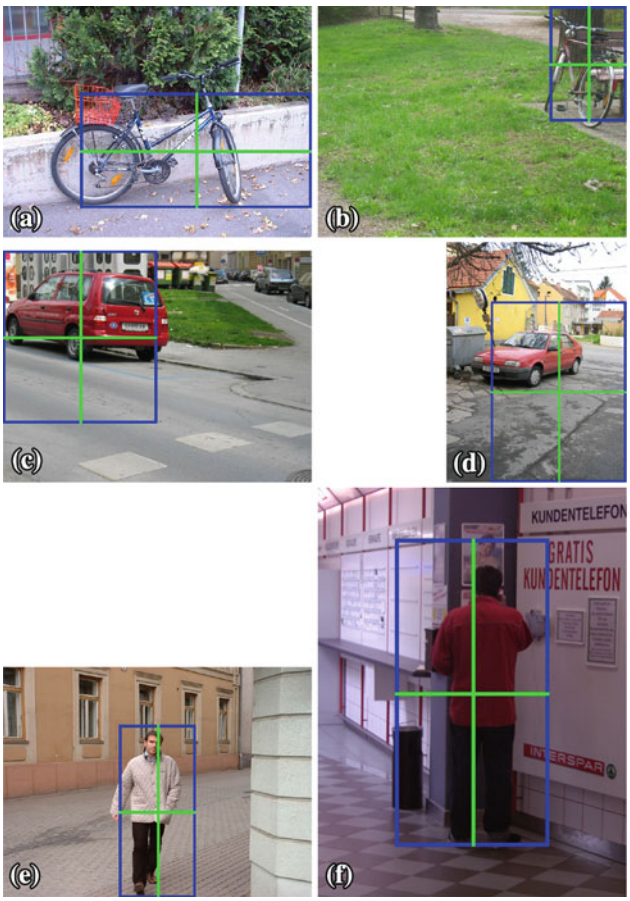


Fig. 5 Representative crop-uniform split examples from the Graz-02 data set. (**a**) and (**b**) show coarse localization of 'bike' images with uniform splitting. (**c**) and (**d**) examples include 'cars' and 'road' in the upper and bottom subdivisions respectively. Differently from the strict bounding box concept in object detection tasks, the inferred image windows contain additional context information. Crop-uniform split achieves a coarse localization of 'person' in different (outdoor and indoor) environments in (**e**) and (**f**) respectively

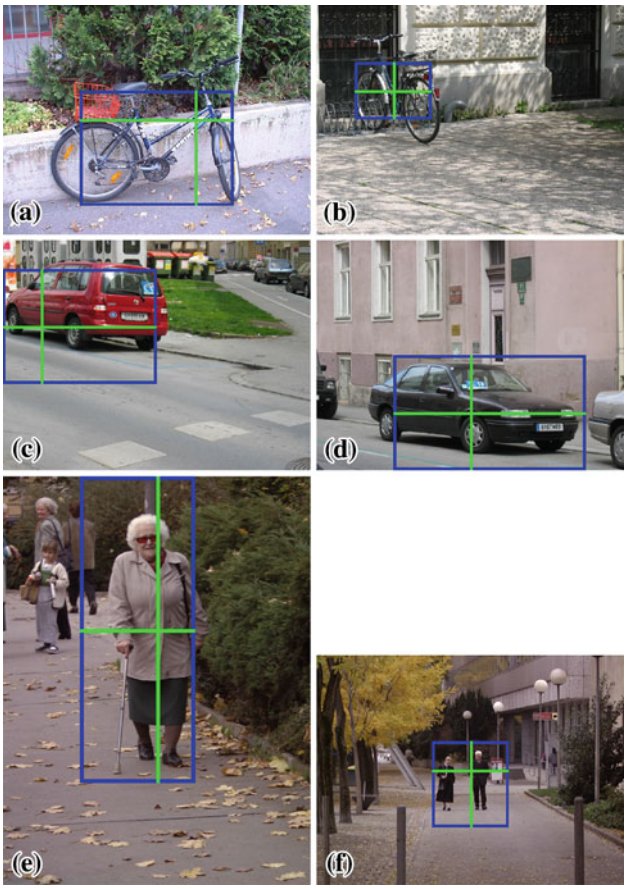


Fig. 6 Representative crop–split examples from the Graz-02 data set. The crop–split is the most flexible operation and it can localize objects and align object parts better than the crop-uniform operation. The advantage of the crop–split over the crop-uni-split can be observed by comparing (**a**) to Fig. 5a. The crop–split achieves better elimination of the background in the image (**a**). In case of multiple objects, it picks the bigger person over the small ones in background in (**e**). The image window in (**f**) contains two people that have similar sizes and are close to each other

ied here. It learns both a cropping box and a non-uniform subdivision thereof. Its latent parameter set is a combination of the Cropping and Splitting operations, $h_{\text{crop+split}} = \{x_0, y_0, x_1, y_1, x_2, y_2\}$ for images and $h_{\text{crop+split}} = \{t_0, t_1, t_2\}$. The effect is illustrated in Figs. 1d and 2d resp. We illustrate crop-split examples with blue cropping boxes and green splits in Fig. 6. This figure already suggests that the crop-split model is able to roughly locate objects, although we do not use any ground truth bounding box locations in training.

3 Inference and Learning

3.1 Inference

In the sequel, we closely follow the notation proposed by Yu and Joachims (2009). The inference problem corresponds to finding a prediction rule that infers a class label y and a set of latent parameters h for a previously unseen image. Formally speaking, the prediction rule $g_w(x)$ maximizes a function $f_w(x, y, h)$ over y and h given the parameter vector w and the image x, where $f_w(x, y, h)$ is the discriminant function that measures the matching quality between input, output and latent parameters:

$$f_w(x, y, h) = w \cdot \psi(x, y, h) \tag{1}$$

where $\psi(x, y, h)$ is a joint feature vector. We use different ψ vectors for multi-class and binary classification tasks. The feature vector for multi-class setting is

$$\psi_{\text{multi}}(x, y, h) = (0^D \ \ldots \ 0^D \ \ \varphi(x, h) \ 0^D \ \ldots \ 0^D)^T \tag{2}$$

where $y \in \{1, \ldots, k\}$ and $\varphi(x, h) \in \mathcal{R}^D$ is a histogram of quantized features, given a latent parameter set, e.g. h_{crop} or h_{split}. 0^D denotes D-dimensional zero row vector. $\varphi(x, h)$ is stacked into position $y \times D$.

The feature vector for binary-class setting is

$$\psi_{\text{bin}}(x, y, h) = \begin{cases} \phi(x, h) = (\varphi(x, h) 0^D)^T, & \text{if } y = 1 \\ -\phi(x) = (0^D - \varphi(x))^T, & \text{if } y = -1 \end{cases} \tag{3}$$

where $y \in \{-1, 1\}$ ($y = 1$ meaning the class is present in the image and $y = -1$ it is not) and $\varphi(x)$ is the feature representation for whole image. While ψ_{multi} is $K \times D$ dimensional (K denotes the number of classes), ψ_{bin} is $2 \times D$. Differently from the multi-class case, we learn to localize only in positive images and fix the image window to whole image to represent negative images for the binary case. However, this is not the only possible representation, one can also localize in negative images similarly to positive images or set all the elements of feature vector of negative images to zero as in Zhu et al. (2010).

The prediction rule g_w can be obtained by maximizing the discriminant function over label and latent space:

$$g_w(x) = \underset{\hat{y} \in \mathcal{Y}, \hat{h} \in \mathcal{H}}{\arg\max} \ f_w(x, \hat{y}, \hat{h}). \tag{4}$$

3.2 Learning

Suppose we are given a set of training samples $X = \{x_1, \ldots, x_n\}$ and their labels $Y = \{y_1, \ldots, y_n\}$ and we want to learn a SVM model w to predict the class label of an unseen example. We also use latent parameters $H = \{h_1, \ldots, h_n\}$ to select the cropping and/or splitting operations that add spatial information to the classifier, as introduced in Sect. 2. In cases where the set of spatial parameters h_i is also specified in the training set (as with training for detection), the standard structural SVM (Tsochantaridis et al. 2004) solves the following optimization problem:

$$\min_w \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\Big[\max_{\hat{y}_i,\hat{h}_i}\big[w\cdot\psi(x_i,\hat{y}_i,\hat{h}_i) + \Delta(y_i,\hat{y}_i,h_i,\hat{h}_i)\big] - w\cdot\psi(x_i,y_i,h_i)\Big] \tag{5}$$

where C is the penalty parameter and $\Delta(y_i,\hat{y}_i,h_i,\hat{h}_i)$ is the loss function. Note that when h_i is given for training set, one can use a single symbol (s_i) to represent both (y_i,h_i).

For the case of classification, the latent variables will typically not come with the training samples however, and need to be treated as latent parameters. To solve the optimization problem in (5) without the labeled windows, we follow the latent SVM formulation of Yu and Joachims (2009):

$$\min_w \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\Big[\max_{\hat{y}_i,\hat{h}_i}\big[w\cdot\psi(x_i,\hat{y}_i,\hat{h}_i) + \Delta(y_i,\hat{y}_i,\hat{h}_i)\big] - \max_{\hat{h}_i}\big[w\cdot\psi(x_i,y_i,\hat{h}_i)\big]\Big] \tag{6}$$

Note that we remove h_i from Δ since it is not given. In the multi-class classification task, we use the 0-1 loss which is $\Delta(y_i,\hat{y}_i,\hat{h}_i) = 1$ if $\hat{y}_i \neq y_i$, and else 0. We will explain the loss function that is designed for binary classification in Sect. 4.

The latent SVM formulation can be rewritten as the difference of two convex functions:

$$\min_w \underbrace{\Big[\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\max_{\hat{y}_i,\hat{h}_i} w\cdot\psi(x_i,\hat{y}_i,\hat{h}_i) + \Delta(y_i,\hat{y}_i,\hat{h}_i)\Big]}_{p(w)} - \underbrace{\Big[C\sum_{i=1}^{n}\max_{\hat{h}_i}\big[w\cdot\psi(x_i,y_i,\hat{h}_i)\big]\Big]}_{q(w)} \tag{7}$$

The difference of those two functions, $p(w) - q(w)$ can be solved by using the Concave–Convex Procedure (CCCP) (Yuille and Rangarajan 2003), where p and q are convex. The generic CCCP algorithm is guaranteed to decrease the objective function (7) at each iteration t and to converge to a local minimum and or a saddle point. In Sect. 5 we suggest an iterative method for avoiding an undesired local minimum and saddle point in the first iterations. The CCCP algorithm to minimize the difference of two convex functions works as follows:

3.3 Algorithm

Initialize $t = 0$ and w_0.

Iterate:

1. Compute hyperplane v_t such that $-q(w) \leq -q(w_t) + (w - w_t)\cdot v_t$ for all w.
2. Solve $w_{t+1} = \arg\min_w p(w) + w\cdot v_t$

We iterate until the stopping condition $[p(w_t) - q(w_t)] - [p(w_{t-1}) - q(w_{t-1})] < \epsilon$. Note that t is typically small (10–100). The first step involves the latent parameter inference problem

$$h_i^* = \arg\max_{\hat{h}_i\in\mathcal{H}} w_t\cdot\psi(x_i,y_i,\hat{h}_i). \tag{8}$$

Computing the new w_{t+1} in the second line involves solving the standard Structural SVM problem (Tsochantaridis et al. 2004) with the inferred latent variables h_i^*:

$$\min_w \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\max_{\hat{y}_i,\hat{h}_i}\big[w\cdot\psi(x_i,\hat{y}_i,\hat{h}_i) + \Delta(y_i,\hat{y}_i,\hat{h}_i)\big] - C\sum_{i=1}^{n}\big[w\cdot\psi(x_i,y_i,h_i^*)\big] \tag{9}$$

Solving the formula (9) requires to compute the constraint

$$\{y_i^*,h_i^*\} = \arg\max_{\hat{y}_i,\hat{h}_i}\big[w\cdot\psi(x_i,\hat{y}_i,\hat{h}_i) + \Delta(y_i,\hat{y}_i,\hat{h}_i)\big] \tag{10}$$

for each sample. This term is called *most violated constraint* in Tsochantaridis et al. (2004) or *loss augmented inference* in Taskar et al. (2005). It corresponds to the most confusing response from another than the actual class or another latent parameter than the inferred one.

4 Optimizing AUC

Multi-class classification performances are typically measured in terms of accuracy, e.g. correctly classified images over total number of images. While this evaluation criterion is informative in the multi-class setting, it can be misleading in binary classification, as the number of positive and negative images are unbalanced. This imbalance increases a lot more in the case of latent window parameters as we deal with more negative samples (all other bounding boxes in an image are considered negative). The area under the ROC curve (AUC), average precision (AP) and precision at fixed recall give a more intuitive and sensitive evaluation in this case.

We evaluate our proposed classifiers in Sect. 6 on various benchmarks including the PASCAL VOC 2007 data set (Everingham et al. 2007) which uses the AP to judge the classification performances. While it is possible to train our classifiers on the basis of accuracy loss and then report testing performance using the AP, Joachims (2005) shows that such difference may result in a suboptimal performance. To the best of our knowledge, there is no prior work which optimizes a structural SVM with latent parameters based on the exact AP measure. However, it is shown that it is possible to optimize a classifier based on the approximated AP

with the Structural SVM (Yue et al. 2007) or to factorize the optimization problem based on dual decomposition (Ranjbar et al. 2012), optimizing both the classifier and the latent parameters with a Structural SVM proved difficult. Therefore, we will train our classifiers using the AUC criterion, which optimizes for a ranking between positive and negative samples similar to the AP and helps to improve performance even when testing on AP. The proposed learning algorithm does not require any extra parameter to weight negative samples, does not worsen computational complexity compared to training on the basis of accuracy loss, and does improve the classification performance. We report our results on the PASCAL VOC 2007 data set and compare the AUC optimized classifiers to the accuracy based baselines in Sect. 6.

The area under the ROC curve can be computed from the number of positive and negative pairs which are ranked in the wrong order, i.e.:

$$\text{AUC} = 1 - \frac{|\text{Swapped Pairs}|}{n^+ \cdot n^-} \tag{11}$$

where n^+ and n^- are the number of positive and negative samples respectively and $\text{Swapped Pairs} = \left\{(i,j) : y_i > y_j \wedge r(x_i) < r(x_j)\right\}$ with a ranking function $(r(x))$. We design the ranking function $(r(x))$ based on the binary representation in (3) as the maximum response for $\psi_{\text{bin}}(x, 1, h) - \psi_{\text{bin}}(x, -1, h)$:

$$r(x) = \max_{\hat{h}} w \cdot (\phi(x, \hat{h}) + \phi(x)) \tag{12}$$

Using the ranking function (12), we can rewrite the swapped pairs that are used to compute the AUC as

$$\begin{aligned}\text{Swapped Pairs} = \Big\{(i,j) :\ & y_i = 1, y_j = -1 \text{ and} \\ & \max_{\hat{h}_{ij}} w \cdot [\phi(x_i, \hat{h}_{ij}) + \phi(x_i)] < \\ & \max_{\hat{h}_{ij}} w \cdot [\phi(x_j, \hat{h}_{ij}) + \phi(x_j)]\Big\}.\end{aligned} \tag{13}$$

where $\hat{h}_{ij}$ denotes the best latent parameter for image x_i on the left hand side and for image x_j on the right hand side respectively.

In order to incorporate the ranking to the latent structural SVM problem, we design the feature vector ψ by substituting individual samples x with positive–negative pairs $\tilde{x}$:

$$\psi(\tilde{x}_{ij}, \tilde{y}_{ij}, \tilde{h}_{ij}) = \begin{cases} \phi(x_i, \tilde{h}_{ij}) - \phi(x_j), & \text{if } \tilde{y}_{ij} = 1 \\ \phi(x_j, \tilde{h}_{ij}) - \phi(x_i), & \text{if } \tilde{y}_{ij} = -1 \end{cases} \tag{14}$$

where $\tilde{x}_{ij} = (x_i, x_j)$ and $\tilde{y}_{ij} = \begin{cases} 1, & \text{if } y_i = 1, y_j = -1 \\ -1, & \text{if } y_i = -1, y_j = 1 \end{cases}$.

Given the label pair $\tilde{y}_{ij}$, $\tilde{h}_{ij}$ denotes a latent parameter for image x_i when $(\tilde{y}_{ij} = 1)$ or for image x_j when $(\tilde{y}_{ij} = -1)$ respectively. Please note that we discard positive–positive and negative–negative pairs in our training, since the AUC is only related to the ranking between positive and negative samples.

The error between the ground truth label set $\tilde{Y} = \{1, \cdots, 1\}$ and the prediction $\hat{\tilde{Y}} = \{\hat{\tilde{y}}_{ij}\}$ is proportional to $(1 - \text{AUC})$ of the original X and Y where $X = \{x_1, \cdots, x_n\}$ and $Y = \{y_1, \cdots, y_n\}$.

$$\Delta_{\text{AUC}}(\tilde{Y}, \hat{\tilde{Y}}) = \sum_{i=1}^{n^+} \sum_{j=1}^{n^-} \frac{1}{2}(1 - \hat{\tilde{y}}_{ij}) \tag{15}$$

Since the loss function in (15) decomposes linearly over the pairwise relationship (y_i, y_j), the most violated constraint $(\tilde{y}^*_{ij}, \tilde{h}^*_{ij})$ can be computed for each pair individually:

$$\sum_{i=1}^{n^+} \sum_{j=1}^{n^-} \arg\max_{\hat{\tilde{y}}_{ij}, \hat{h}_{ij}} \ w \cdot \psi(\tilde{x}_{ij}, \hat{\tilde{y}}_{ij}, \hat{\tilde{h}}_{ij}) + \frac{1}{2}(1 - \hat{\tilde{y}}_{ij}). \tag{16}$$

The most violated constraint computation for a given image pair $\tilde{x}_{ij} = (x_i, x_j)$ and corresponding label $y_{ij} = 1$ requires to check the inequality:

$$\begin{aligned} &\max_{\hat{h}_{ij}} w \cdot \Big[\phi(x_i, \hat{h}_{ij}) + \phi(x_i)\Big] < \\ &\max_{\hat{h}_{ij}} w \cdot \Big[\phi(x_j, \hat{h}_{ij}) + \phi(x_j)\Big] + 1 \end{aligned} \tag{17}$$

On the other hand, using the accuracy (0-1) loss and the feature representation in (3) leads to the following constraint computation which only considers responses from individual samples:

$$\begin{aligned} &\max_{\hat{h}_i} w \cdot \phi(x_i, \hat{h}_i) < -w \cdot \phi(x_i) + 1, \ \text{if } y_i = 1 \\ &-w \cdot \phi(x_i) < \max_{\hat{h}_i} w \cdot \phi(x_i, \hat{h}_i) + 1, \ \text{if } y_i = -1. \end{aligned} \tag{18}$$

In practice, computing (17) for each pair does not add any significant computation load since $\max_{\hat{h}_i}(w \cdot \phi(x_i, \hat{h}_i))$ and $(w \cdot \phi(x_i))$ can be precomputed for each sample (x_i) individually.

We can now write the latent SVM formulation in (7) for the AUC optimization. To do so, we define the convex functions $p(w)$ and $q(w)$ for brevity, and their difference can be used to compute the complete formulation. $p(w)$ is written as sum of a regularization term and (16):

$$\begin{aligned} p(w) = \frac{1}{2}\|w\|^2 + C\Bigg[&\sum_{i=1}^{n^+} \sum_{j=1}^{n^-} \max_{\hat{\tilde{y}}_{ij}, \hat{h}_{ij}} \ w \cdot \psi(\tilde{x}_{ij}, \hat{\tilde{y}}_{ij}, \hat{\tilde{h}}_{ij}) \\ &+ \frac{1}{2}(1 - \hat{\tilde{y}}_{ij})\Bigg]. \end{aligned} \tag{19}$$

In contrast to $p(w)$, the second convex function $q(w)$ can be computed linearly in terms of individual samples (x) by using the feature representation (14):

$$q(w) = C\left[n^- \sum_{\substack{i,\\ y_i=1}} \max_{\hat{h}_i} w \cdot \phi(x_i, \hat{h}_i) - n^+ \sum_{\substack{j,\\ y_j=-1}} w \cdot \phi(x_j)\right]. \tag{20}$$

So far, we have detailed the learning procedure that makes use of positive–negative image pairs (x_i, x_j) and penalizes ranking violations between those pairs. In parallel to the learning procedure, the prediction rule ranks images by using (12). The inference for an unseen image is rewritten as

$$g_{\text{AUC}}(x) = \begin{cases} y^* = 1, \text{if } \max_{\hat{h}} w \cdot (\phi(x, \hat{h}) + \phi(x)) > 0 \\ y^* = -1, \text{else.} \end{cases} \tag{21}$$

5 Iterative Learning of Latent Parameters

Learning the parameters of an LSVM model often requires solving a non-convex optimization problem. Like every such problem, LSVM is also prone to getting stuck in local minima. Recent work (Bengio et al. 2009) proposes an iterative approach to find better local minima within shorter convergence times for non-convex optimization problems. It suggests to first train the learning algorithm with easy examples and to then gradually feed in more complex examples. This procedure is called curriculum learning. The main challenge of curriculum learning is to find a good measure to quantify the difficulty of samples.

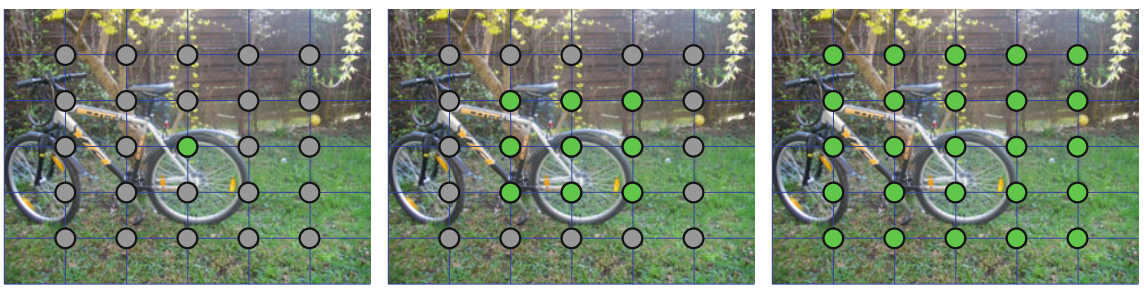
In this paper, we take the size of the parameter space as an indication of the complexity of the learning problem. Initially, we run the learning algorithm with a limited latent subspace and then gradually increase the latent parameter space. Figure 7 illustrates such iterative learning for the splitting operation. The nodes located in the corners of the grid indicate the possible splitting points, i.e. the latent parameter set for the splitting operation. The green nodes indicate, from left to right, the growing number of splitting points that the algorithm can choose from during subsequent iterations.

6 Experiments

We evaluate our system on four publicly available computer vision benchmarks, the Graz-02 (Opelt et al. 2006), the PASCAL VOC 2007 (Everingham et al. 2007) and the Caltech 101 (Fei-Fei et al. 2004) data sets for object classification, and the activities of daily living life data set (Messing et al. 2009) for action classification.

For the object classification experiments, we extract dense SIFT features (Lowe 1999) by using the `vl_phow` function from the VLFeat toolbox (Vedaldi and Fulkerson 2008). For the action classification experiments, we use the HoF descriptors (Laptev et al. 2008) to describe detected Harris3D interest points (Laptev and Lindeberg 2003). We apply K-means to the randomly sampled 200,000 descriptors from the training images/videos to form the visual codebook. The computed visual words are then used to encode the descriptors with the LLC method (Wang et al. 2010). For the LLC encoding, we set the number of nearest neighbors and the regularization parameter to 5 and 10^{-4} respectively. The codebook sizes are 1024, 8192, 2048 and 1000 for the Graz-02, VOC-07, Caltech-101 and the Activities data sets respectively (often following the sizes used by others, in order to allow for a fair comparison in the subsequent experiments).

We compare the performance of the proposed latent operations, 'crop', 'split', 'crop-uni-split', 'crop–split' to the standard *bag-of-features* (BoF) and one level spatial pyramid (SP) (Lazebnik et al. 2006). The BoF represents an image/video with a histogram of quantized local features and thus discards the spatial/temporal layout of the image/video structure. The SP is a more extensive representation which incorporates spatial information into the features by using a pyramidal representation. In our experiments, we use a one level SP (1×1 for the top layer and 2×2 for the base) for images, and a similar SP for videos, where the base is only temporally divided. The performance criterion is the mean multi-class classification accuracy for the Graz-02,



(a) `iter 0` **(b)** `iter 1` **(c)** `iter 2`

Fig. 7 Illustration of the splitting operation in iterative learning. The *green* and *gray* nodes show the points where splitting is considered. At `iter 0` the image can only be splitted with *horizontal* and *vertical* lines through the image center, while at the next iteration `iter1`, the image can be splitted with one of the 9 *green* nodes. At the last iteration `iter2`, all splitting nodes are eligible (Color figure online)

Caltech-101 and the Activities data sets and mean AP (mAP) for the VOC-07. Similarly, the feature representation of the 'split', 'crop-uni-split' and 'crop–split' operations are equal with the SP.

Our latent learning implementation builds on the publicly available code of Yu and Joachims (2009). The regularizing parameter C of the LSVM is tuned for each latent operation (crop, split, etc.) on each data set (Graz, VOC-07, etc.) by using cross-validation (the interval $[10^2, 10^7]$ is sampled logarithmically). The other free parameter ϵ, the stopping criterion for the CCCP algorithm, is set to 10^{-1} and 10^{-3} for the multi-class and binary classification experiments, respectively.

The running time of the LSVM experiments is dominated by computing the 'most violated constraint' which was introduced in Sect. 4. We need to compute the response of each classifier by scanning the latent parameter space (e.g. all possible boxes for the cropping operation), to find the violated constraints. It would therefore have been possible to improve the running time by using the branch and bound algorithm (Lampert et al. 2008). For the cropping, splitting, crop-uniform-splitting, and crop-splitting operations the training of each class-specific classifier in the VOC 2007 experiments took 1 h, 5 min, 30 min and 3 h on a 16 CPU machine, resp. Training for the other data sets went faster, and in the same relative orders of magnitude for the different operations.

6.1 Graz-02 Dataset

The Graz-02 data set contains 1096 natural real-world images with three object classes: bikes, cars and people. This database includes a considerable amount of intra-class variation, varying illumination, occlusion, and clutter. We form 10 training and testing sets by randomly sampling 150 images from each object class for training and use the rest for testing. We report the mean and standard deviation of the classification accuracy for the 10 corresponding experiments, each time also averaging over the 3 classes.

Table 1 shows the multi-class classification results. The crop operation improves the classification performance over the BoF and the SP representation by around 1.45 and 0.35 %, respectively. The non-uniform split operation also achieves better classification performance than the uniform split (SP). The crop-split operation has more degrees of freedom than the crop-uni-split model and outperforms the crop-uni-split: where the latter improves the baseline SP method by 2.4 %, the former improves it by 2.6 %. The crop-split operation thereby also gives the best result of all four operations. Adding splits systematically improved results over pure crops. This may not come as a surprise, as our implementation of splitting leads to substantially larger feature spaces (as SP does compared to BoF).

For cropping and splitting, we only consider points that lie on a regular grid. We now analyze the influence of the size of this grid on the classification accuracy. Figure 8 plots the mean classification accuracy of the four proposed operations for the Graz-02 data set, and this for different grid sizes, i.e. 4×4, 8×8, 12×12, and 16×16. The results show that the performance of the classifiers increases with finer grids up to size 12, after which it slightly drops at 16. Hence, the optimal grid size on the Graz-02 data set is 12. Note that an increased grid size implies a significant, about quadratic, increase in computation time. We therefore report results for all other data sets with a grid size of 8.

6.2 PASCAL VOC 2007

The PASCAL VOC 2007 data set (Everingham et al. 2007) (VOC-07) contains 9,963 images which are split into training, validation and testing sets. The images are labeled with twenty classes, also allowing multiple classes to be present in the same image. We learn a one-vs-rest classifier for each class and report the mean average precision (mAP) which is the mean of AP values from each of the classifiers.

Table 1 depicts the classification results for the proposed operations. It should be noted that we use the AUC-loss based optimized classifiers for both the baseline and proposed latent

Table 1 The classification results on the Graz-02, PASCAL VOC 2007, Caltech-101 and the activities of daily living data set

Dataset	Baseline		Our work			
	BoF	SP	Crop	Split	Crop-uni-split	Crop–split
Graz-02	86.95 ± 1.35	88.05 ± 1.39	88.40 ± 1.05	88.58 ± 1.31	90.38 ± 1.85	**90.62 ± 1.75**
VOC-07	49.86	54.74	51.82	55.32	56.26	**57.05**
Caltech 101	61.25 ± 0.88	72.68 ± 1.21	62.16 ± 0.96	73.33 ± 0.98	**75.31 ± 0.68**	74.93 ± 0.86
Activities	79.33	88.00	72.00	88.00	**90.67**	88.67

The performance of the crop, split, crop-uniform split and crop–split operations are compared to the baselines: BoF and SP. All the classifiers are learnt with the iterative LSVM. We use the AUC based optimization to train the baseline and proposed classifiers for the VOC-07 data set

Bold values indicate the best result among all the methods

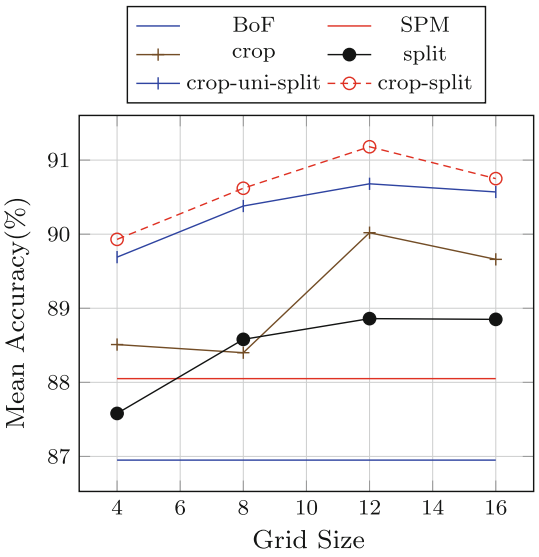


Fig. 8 The mean classification accuracy on the Graz-02 data set with varying grid size. The grid size of 12 gives the best score for the crop, split, crop-uni-split and crop–split operations.

operations to present a fair comparison. The 'crop' operation yields an improvement of around 2 % over the baseline BoF method to which it is similar in terms of feature space dimension. The 'split' operation improves the result over the SP method by 0.6 %. The latent operations of 'crop-uni-split' and 'crop–split' provide further improvements over the SP and BoF baselines. Compared to SP, the 'crop-uni-split' operation yields an improvement of 1.5 % and 'crop–split' one of 2.3 %.

Table 2 shows the results for each object class individually for the crop-split operation. As can be observed from the results, we are able to improve the classification accuracy for 17 out 20 classes. In particular, the crop-split achieves substantial improvement in 'bus' (5.1 %), 'sofa' (5.0 %), 'bicycle' (4.5 %), 'motorbike' (4.3 %) and 'tv monitor' (4 %) categories. The method is not able to improve the accuracy for classes that are hard to localize because of their relatively size and cluttered background around them, such as 'bottle' and 'potted plant'.

6.3 Caltech-101 Dataset

The Caltech-101 data set (Fei-Fei et al. 2004) contains images of 101 object classes and an additional background class, i.e. 102 classes in total. The number of images per class varies from 31 to 800. We use 30 images for training from each class and use the rest of the images—as usual with a maximum number of 50—for testing. We run ten experiments on ten random divisions between training and testing images and report the mean accuracy and standard deviation for these runs.

Table 1 depicts the classification results for the Caltech 101 data set. The crop and split operations improve over the BoF and SP baselines respectively as in the previous data sets. For this data set, where objects are always centered, the crop-uni-split operation achieves the highest performance among the proposed methods and improves the SP method by around 2.6 %.

6.4 The Activities of Daily Living Dataset

The Activities data set (Messing et al. 2009) contains ten different types of complex actions like answering a phone, writing a phone number on a white-board and eating food with silverware. These activities are performed three times by five people with different heights, genders, and ethnicities. Videos are taken at high resolution (1280 × 720 pixels). A leave-one-out strategy is used for all subjects and the results are averaged as in Messing et al. (2009).

Table 1 shows the results for action classification on this data set. For this method, we obtain an improvement of 2.6 % over SP method using the 'crop-uni-split' method. This is similar to the performance for classification of objects and indicates that the method is applicable to the classification of actions as well. The decrease in results for the 'crop' operation over the BoF method is mainly due to the fact that the HOF descriptors are not densely computed and some temporal cells of the grid have very few descriptors.

Table 2 The classification results in terms of AP for each class of PASCAL VOC 2007

Method	mAP	Plane	Bicycle	Bird	Boat	Bottle	Bus	Car	Cat	Chair	Cow
SP	54.74	69.95	59.62	45.42	64.39	**24.81**	60.43	75.31	57.45	53.48	42.87
Crop–split	**57.05**	**72.76**	**64.15**	**46.10**	**66.49**	24.22	**65.57**	**78.64**	**60.55**	**55.02**	**44.23**
		Table	Dog	Horse	mbike	Person	Plant	Sheep	Sofa	Train	TV
SP		46.90	**41.23**	71.38	62.70	82.44	**22.46**	43.54	49.58	70.92	49.99
Crop–split		**48.70**	41.01	**73.33**	**67.05**	**83.93**	21.38	**46.28**	**54.56**	**72.91**	**54.06**

Both the SP and crop–split classifiers are trained with the iterative learning and AUC loss. The crop-split operation out-performs the SP in 17 out of 20 classes and the average improvement is 2.3 % mAP

Bold values indicate the best result among all the methods

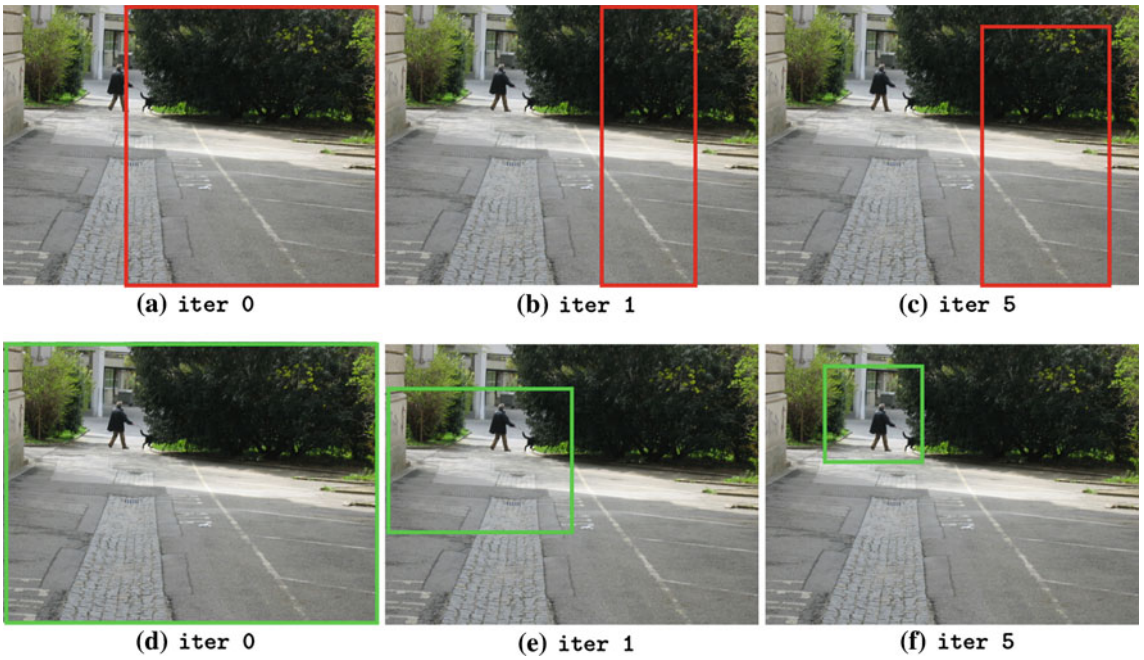


Fig. 9 Cropping operation on a 'person' labeled image for various iterations during training. The first and second rows show the result of the ordinary and iterative learning respectively. The first learning algorithm misses the 'person' in the first iteration and later converges to some part of background. The same local minimum is avoided in the second learning algorithm by restricting the possible image windows set to the full image in the first iteration and gradually relaxing the restriction

Table 3 Comparison of the LSVM and Iterative LSVM in terms of the multi-class classification accuracy for the proposed latent operations on the Graz-02 data set

	Crop	Split	Crop-uni-split	Crop–split
LSVM	89.91 ± 1.69	**88.91 ± 1.37**	90.37 ± 1.21	90.32 ± 1.69
Iter. LSVM	**90.02 ± 1.37**	88.86 ± 1.05	**90.68 ± 1.24**	**91.18 ± 1.38**

Bold values indicate the best result among all the methods

6.5 Iterative Learning

We show results for the iterative learning of latent operations on the Graz-02, VOC-07 and Caltech-101 data sets. The grid size used for the Graz-02 data set is 12×12 and 8×8 for the VOC-07 and Caltech-101 data sets. For the split operation we initially constrain the latent search space to the center of the images and expand it along the x and y directions by a fixed step size, a quarter of the number of rows and columns in the grid, e.g. $12/4 = 3$ on the 12×12 grid, at each iteration. For the crop, crop-uni-split, and crop-split operations, we initially fix the image window, e.g. $\{x_1, y_1, x_2, y_2\}$, as the full image. At each iteration, we relax the minimum width and height of the image window with a fixed step size, i.e. $0.5 \times$ grid size. Once the CCCP algorithm converges within the given latent space in an iteration, we expand the latent search space again at the start of the next. The algorithm terminates when the entire search space is covered.

Figure 9 visualizes key iterations of the training for the cropping operation of a 'person' image for the LSVM and iterative LSVM. In the iterative scheme, we initially fix the latent cropping box to be the full image size at the `iter 0` (Fig. 9a). We then relax the constraint by allowing a smaller minimum size of the cropping box, i.e. half of the minimum size from the previous iteration. The ordinary LSVM method does not have any such constraint on the latent parameter search. At the end of `iter 0`, the LSVM converges to a wrong region and the error propagates to the next iterations. The LSVM mis-classifies this training image as 'bike'. The iterative LSVM gradually learns to localize the person better and correctly classifies the image.

Table 3 depicts the quantitative result of the iterative operations on the Graz-02 data set. The table indicates that the iterative method for LSVM generally improves the classification accuracy over the original formulation of the LSVM. The crop-split benefits most from the iterative method, since it has more degrees of freedom and thus a stronger tendency

Table 4 Comparison of the LSVM and iterative LSVM on different data sets for the crop–split operation

	Graz-02	VOC-07	Caltech101
LSVM	90.32 ± 1.69	56.00	**75.04 ± 0.76**
iter. LSVM	**91.18 ± 1.38**	**57.05**	74.93 ± 0.86

Iterative LSVM performs better in both the Graz-02 and VOC-07 data sets. The Caltech-101 data set does not benefit from the iterative method, since the images in this data set do not contain significant background clutter. Therefore, image windows are not less likely to converge to non-representative image parts in this data set

Bold values indicate the best result among all the methods

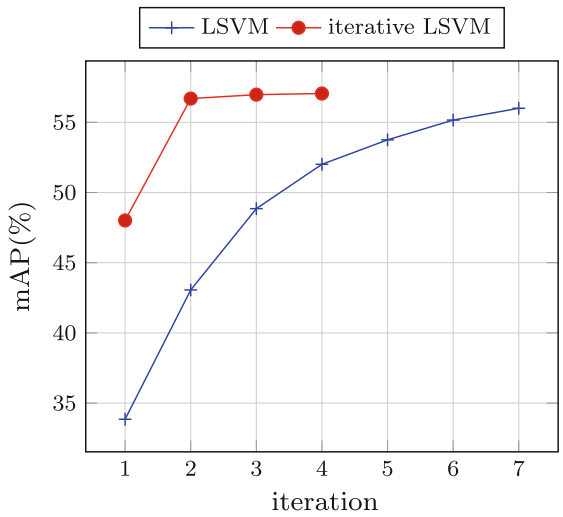


Fig. 10 Classification results (mAP) with the AUC optimized crop–split on the VOC-07 over iterations for LSVM and iter LSVM algorithms. The minimum image windows size is limited to whole image size and half of it during the first and second iterations of the iterative learning respectively. The iterative learning starts with higher classification mAP on testing and takes fewer iterations to converge. The LSVM and iter LSVM converge to 56 and 57.05 % mAP respectively.

to converge to a local minimum. The performance of iterative learning for the split operation worsens slightly.

Table 4 shows quantitative comparison of iterative learning for the crop-split operation on the Graz-02, VOC-07 and Caltech-101 data sets. The iterative learning improves the classification performance for the Graz-02 and VOC-07 around 1 %. However, we observe a slight drop in the classification accuracy on the Caltech-101. In the Caltech-101 data set objects are well centered, objects do not vary significantly in their sizes and the images are quite clean of clutter. Therefore, this data set does not benefit from the proposed learning method.

Figure 10 plots the classification performance of the LSVM and iter LSVM for the crop-split operation on the VOC-07 data set over iterations. The CCCP algorithm, as described in Sect. 3.3, at beginning of each iteration, infers the latent variables. Having the latent parameters fixed, it optimizes the minimization problem 9 during that iteration. We limit the minimum image window size for the iter LSVM to whole and half image size during the first and second iterations respectively. We observe that the iter LSVM already has 48 % mAP at the end of the first iteration and converges fast to 57.05 % mAP. However, the LSVM takes 7 iterations to converge to 56 % mAP.

Table 5 Comparison between the accuracy loss (ACC), normalized accuracy loss (N-ACC) and area under the roc curve loss (AUC) on the VOC-07 data set in mAP

Loss	SP (mAP)	Crop–split (mAP)
ACC	53.46	54.37
N-ACC	54.18	56.98
AUC	**54.57**	**57.05**

Bold values indicate the best result among all the methods

6.6 AUC Optimization

In Sect. 4, we described the use of an AUC based objective function to learn the classification with latent variables. This is useful in the case of binary classification, e.g. the VOC 2007 object classification task. For this task, we compare the proposed AUC loss against two baselines (ACC and N-ACC) in Table 5. ACC denotes the 0-1 or accuracy loss. N-ACC is normalized accuracy loss for the number of positives and negatives, e.g. it penalizes false negatives more in presence of more negative images. We evaluate their performances for the standard SP and latent crop–split operation. While the ACC loss performs worst in all three data sets, normalizing the loss (N-ACC) for positives and negatives with the number of positives and negatives respectively improves the mAP in both SP and crop–split. The AUC loss gives the best results and empirically shows that the AUC loss provide a better approximation of the AP on the VOC-07 data set than the ACC and N-ACC baselines.

6.7 Statistical Significance of Results

In this section, we further analyze whether the difference in performance between the proposed latent operations and the baselines is statistically significant. There is little work in the literature that studies statistical evaluation of multiple classifiers on multiple data sets. We analyze our results by following two different evaluation tests which is recommended by the authors of Demšar (2006).

In the first analysis, we group the methods in terms of their feature dimension to have fair comparison. We explore whether the ‘crop’ operation produce statistically significant difference over the ‘control’ or baseline classifier BoF. We also compare the ‘split’, ’crop-uni-split’ and ‘crop–split’

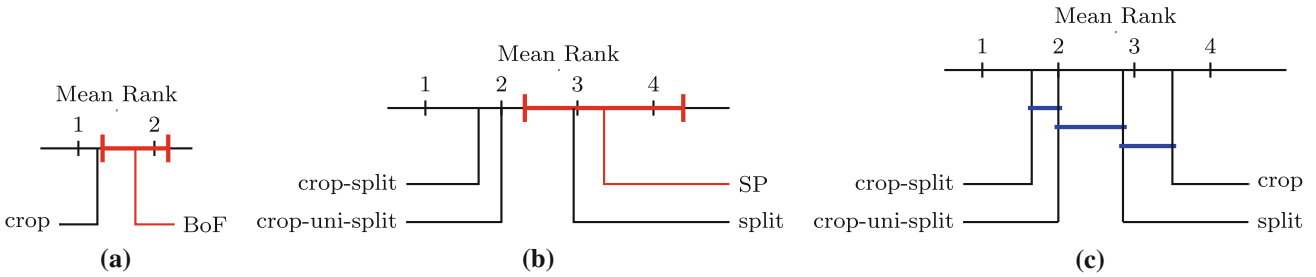


Fig. 11 Significance analysis of the classification results on the VOC-07 data set. (**a**) Shows a comparison of the BoF against the crop operation with the Bonferroni–Dunn test. The crop operation is outside the marked *red* interval is significantly different ($p < 0.05$) from the control classifier BoF. (**b**) Shows comparison of the SPM against the split, crop-uni-split and crop–split operations with the Bonferroni–Dunn test. While the crop-uni-split and crop–split operations are outside of the *red* marked range, therefore they are significantly better ($p < 0.05$) than SP. (**c**) Shows comparison of all the proposed latent operations against each other with the Nemenyi test. Groups of classifiers that are not significantly different (at $p < 0.05$) are connected (Color figure online)

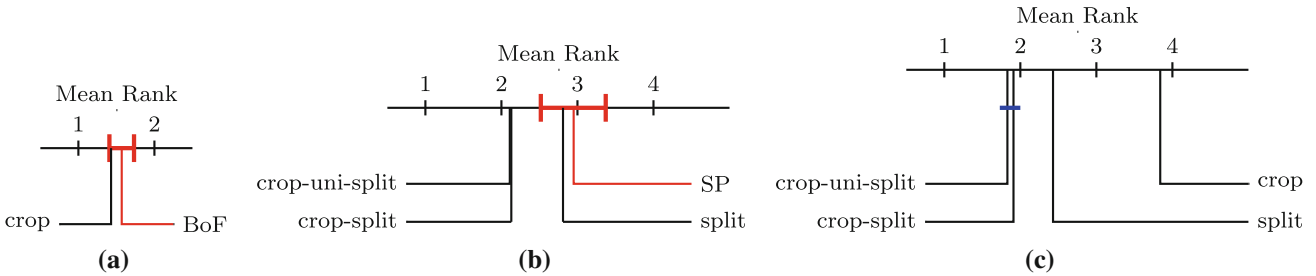


Fig. 12 Significance analysis of the classification results on the Caltech-101 data set. (**a**) Shows a comparison of the BoF to the crop operation with the Bonferroni–Dunn test. The crop operation is inside the *red* marked interval is not significantly different ($p < 0.05$) from the control classifier BoF. (**b**) Shows comparison of the SPM to the split, crop-uni-split and crop–split operations with the Bonferroni–Dunn test. While the crop-uni-split and crop–split operations are outside of the *red* marked range, therefore they are significantly better ($p < 0.05$) than SP. (**c**) Shows comparison of all the proposed latent operations to each other with the Nemenyi test. Groups of classifiers that are not significantly different (at $p < 0.05$) are connected (Color figure online)

operations to the SP. More specifically, we followed the two step approach of the Friedman test (Friedman 1937) with the Bonferroni–Dunn *post-hoc* analysis (Dunn 1961). This approach ranks the classifiers in terms of their classification results (highest classification accuracy is ranked 1, 2nd one is ranked 2) and therefore it does not require any assumptions about the distribution of the accuracy or AP to be fulfilled. In our experiments, we consider each class as a separate test and rank each class among different methods. We test the hypothesis that it could be possible to improve on the control classifiers (BoF, SP) by using the latent operations. The null hypothesis which states that all the algorithms are equivalent is tested by the Friedman test. After the null hypothesis is rejected, we use the Bonferroni–Dunn test which gives a "critical difference" (CD) to measure the difference in the mean rank of the control and proposed classifiers.

Figures11a, b and 12a, b depict the results of the first analysis for the VOC-07 and Caltech-101 data sets respectively. This diagram is proposed by (Demšar 2006). The top line in the diagrams is the axis which indicates the mean ranks of methods in an ascending order from the lowest (best) to the highest (worst) rank. We mark the interval of CD to the left and right of the mean rank of the control algorithm (BoF and SP) in Figs.11a, b and 12a, b. The algorithms with the mean rank outside this range are significantly different from the control. Figure11a, b depict that the crop performs significantly better than the BoF; crop-uni-split and crop-split are significantly better than the SP on the VOC-07. Figure12a, b show that the crop is not significantly better than the BoF, the crop-uni-split and crop–split are still significantly better than the SP on the Caltech-101. While the VOC-07 data set images include cluttered background and small objects embedded in challenging backgrounds, the Caltech-101 images are cleaner. Therefore, only 'crop' operation cannot perform significantly better than BoF in the latter data set. The 'split' operation has enough degree of freedom to improve over the SP in neither of the data sets.

In the second analysis, we compare the performance of the latent operations to each other. We follow the same testing strategy with the authors of Everingham et al. 2010 to analyze the significance of the results. We have used the Friedman test with a different post hoc test, known as Nemenyi test (Nemenyi 1963). While Bonferroni–Dunn test is more suitable to compare the proposed algorithms with a control

classifier, Nemenyi test is more powerful to compare all classifiers to each other. This test also computes a CD to check whether the difference in mean rank of two classifiers is bigger than this value. We show results of the second analysis for the VOC-07 and Caltech-101 data sets in Figs.11c and 12c respectively. Figure11c shows that the 'crop' and 'split' are not significantly different from each other in terms of their classification performance, however, their combination 'crop-split' is significantly better than both 'crop' and 'split'. This shows that these two operations are different approaches to learn and complementary to each other. In both Figs.11c and 12c the 'crop-uni-split' and 'crop–split' are not significantly different from each other. This is because splitting can only marginally improve the histograms by redistributing features and this results in an improvement, but not a statistically significant improvement of the result.

7 Conclusion and Future Work

We have developed a method for classifying objects and actions with latent window parameters. We have specifically shown that learning latent variables for flexible spatial operations like 'crop' and 'split' are useful for inferring the class label. We have adopted the latent SVM method to jointly learn the latent variables and the class label. The evaluation of our principled approach yielded consistently good results on several standard object and action classification data sets. We have further improved the latent SVM by iteratively growing the latent parameter space to avoid local optima. We also realized a better learning algorithm for unbalanced data by using an AUC based objective function. In the future, we are interested in extending the approach for weakly supervised object detection and improved large scale classification.

References

Bengio, Y., Louradour, J., Collobert, R., & Weston, J. (2009). Curriculum learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, ACM (pp. 41–48).

Bilen, H., Namboodiri, V. P., & Van Gool, L. (2011). Object and action classification with latent variables. In *Proceedings of The British Machine Vision Conference*.

Bilen, H., Namboodiri, V. P., & Van Gool, L. (2012). Classification with global, local and shared features. In *Proceedings of The DAGM-OAGM Conference*.

Blaschko, M.B., Vedaldi, A., & Zisserman, A. (2010). Simultaneous object detection and ranking with weak supervision. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*.

Boureau, Y.L., Bach, F., LeCun, Y., & Ponce, J. (2010). Learning mid-level features for recognition. In: *Proceedings of Computer Vision and Pattern Recognition (CVPR)* (pp. 2559–2566).

Chatfield, K., Lempitsky, V., Vedaldi, A., & Zisserman, A. (2011). The devil is in the details: an evaluation of recent feature encoding methods. In *British Machine Vision Conference*.

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research*, *7*, 1–30.

Dunn, O. (1961). Multiple comparisons among means. *Journal of the American Statistical Association*, *56*(293), 52–64.

Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., & Zisserman, A. (2010). The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, *88*(2), 303–338.

Everingham, M., Zisserman, A., Williams, C.K.I., & Van Gool, L. (2007). The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results.

Fei-Fei, L., Fergus, R., & Perona, P. (2004). Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *IEEE. CVPR 2004, Workshop on Generative-Model Based Vision*.

Felzenszwalb, P. F., Girshick, R. B., McAllester, D., & Ramanan, D. (2010). Object detection with discriminatively trained part based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *32*(9), 1627–1645.

Friedman, M. (1937). The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the American Statistical Association*, *32*(200), 675–701.

Gehler, P.V., & Nowozin, S. (2009). On feature combination for multiclass object classification. In *Proceedings of International Conference on Computer Vision (ICCV)* (pp. 221–228).

Joachims, T. (2005). A support vector method for multivariate performance measures. In *Proceedings of the 22nd international conference on Machine learning, ACM* (pp. 377–384).

Kumar, M.P., Packer, B., & Koller, D. (2010). Self-paced learning for latent variable models. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)* (pp. 1189–1197).

Lampert, C., Blaschko, M., & Hofmann, T. (2008). Beyond sliding windows: Object localization by efficient subwindow search. In *IEEE Conference on Computer Vision and Pattern Recognition, (CVPR) 2008* (pp. 1–8) doi:10.1109/CVPR.2008.4587586

Laptev, I., & Lindeberg, T. (2003). Space-time interest points. In *Proceedings of International Conference on Computer Vision (ICCV)* (pp. 432–439).

Laptev, I., Marszałek, M., Schmid, C., & Rozenfeld, B. (2008). Learning realistic human actions from movies. In *Proceedings of Computer Vision and Pattern Recognition (CVPR)*.

Lazebnik, S., Schmid, C., & Ponce, J. (2006). Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proceedings of Computer Vision and Pattern Recognition (CVPR)* (pp. 2169–2178).

Lowe, D. (1999). Object recognition from local scale-invariant features. In *Proceedings of International Conference on Computer Vision (ICCV)* (p. 1150).

Messing, R., Pal, C., & Kautz, H. (2009). Activity recognition using the velocity histories of tracked keypoints. In *Proceedings of International Conference on Computer Vision (ICCV)*. Washington, DC.

Nemenyi, P. (1963). *Distribution-free multiple comparisons*. Ph.D. Thesis, Princeton.

Nguyen, M. H., Torresani, L., De la Torre, F., & Rother, C. (2009). Weakly supervised discriminative localization and classification: a joint learning process. In *Proceedings of International Conference on Computer Vision*.

Opelt, A., Pinz, A., Fussenegger, M., & Auer, P. (2006). Generic object recognition with boosting. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, *28*(3), 416–431.

Perronnin, F., Sánchez, J., & Mensink, T. (2010). Improving the fisher kernel for large-scale image classification. In *European Conference on Computer Vision (ECCV) (4)* (pp. 143–156).

Pinz, A. (2005). Object categorization. *Foundations and Trends in Computer Graphics and Vision*, *1*(4), 255–353.

Ranjbar, M., Vahdat, A., Mori, G. (2012). Complex loss optimization via dual decomposition. In: Computer Vision and Pattern Recognition (CVPR). *2012 IEEE Conference on*, pp. 2304–2311. IEEE.

Satkin, S., Hebert, M. (2010). Modeling the temporal extent of actions. In *Proceedings of European Conference Computer Vision (ECCV)* (pp. 536–548).

Schüldt, C., Laptev, I., Caputo, B. (2004). Recognizing human actions: A local svm approach. In *International Conference on Pattern Recognition (ICPR)* (pp. 32–36).

Shapovalova, N., Vahdat, A., Cannons, K., Lan, T., Mori, G. (2012). *Similarity constrained latent support vector machine: An application to weakly supervised action classification*. In: Proc. of European Conf. Computer Vision (ECCV).

Sharma, G., Jurie, F., Schmid, C. (2012). Discriminative spatial saliency for image classification. In: *2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 3506–3513).

Taskar, B., Chatalbashev, V., Koller, D., Guestrin, C. (2005). Learning structured prediction models: A large margin approach. In *Proceedings of the 22nd international conference on Machine learning, ACM* (pp. 896–903).

Tsochantaridis, I., Hofmann, T., Joachims, T., Altun, Y. (2004). Support vector machine learning for interdependent and structured output spaces. In *Proceedings of International Conference on Machine Learning (ICML)* (p. 104).

Vedaldi, A., & Fulkerson, B. (2008). VLFeat: An open and portable library of computer vision algorithms. http://www.vlfeat.org/. Accessed 10 Jan 2012.

Vedaldi, A., Gulshan, V., Varma, M., Zisserman, A. (2009). *Multiple kernels for object detection*. In: Proc. of Int. Conf. on Computer Vision (ICCV), pp. 606–613.

Vedaldi, A., & Zisserman, A. (2009). Structured output regression for detection with partial occulsion. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*.

Wang, H., Ullah, M.M., Kläser, A., Laptev, I., Schmid, C. (2009). Evaluation of local spatio-temporal features for action recognition. In *Proceedings of British Machine Vision Conference (BMVC)*, (p. 127).

Wang, J., Yang, J., Yu, K., Lv, F., Huang, T.S., Gong, Y. (2010). Locality-constrained linear coding for image classification. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, (pp. 3360–3367).

Yu, C.N.J., Joachims, T. (2009) Learning structural svms with latent variables. In *Proceedings of International Conference on Machine Learning (ICML)* (pp. 1169–1176).

Yue, Y., Finley, T., Radlinski, F., Joachims, T. (2007) . A support vector method for optimizing average precision. In *ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)* (pp. 271–278)

Yuille, A., & Rangarajan, A. (2003). The concave–convex procedure. *Neural Computation*, *15*(4), 915–936.

Zhou, X., Yu, K., Zhang, T., Huang, T.S. (2010). Image classification using super-vector coding of local image descriptors. In *European Conference on Computer Vision (ECCV) (5)* (pp. 141–154).

Zhu, L., Chen, Y., Yuille, A., Freeman, W. (2010). Latent hierarchical structural learning for object detection. *IEEE Conference on Computer Vision and Pattern Recognition, (CVPR) 2010* (pp. 1062–1069).