Noname manuscript No.
(will be inserted by the editor)

Hierarchical Adaptive Structural SVM for Domain Adaptation

Jiaolong Xu · Sebastian Ramos · David Vázquez · Antonio M. López

Received: date / Accepted: date

Abstract A key topic in *classification* is the accuracy loss produced when the data distribution in the training (*source*) domain differs from that in the testing (*target*) domain. This is being recognized as a very relevant problem for many computer vision tasks such as image classification, object detection, and object category recognition. In this paper, we present a novel *domain adaptation* method that leverages multiple target domains (or sub-domains) in a hierarchical adaptation tree. The core idea is to exploit the commonalities and differences of the jointly considered target domains.

Given the relevance of structural SVM (SSVM) classifiers, we apply our idea to the adaptive SSVM (A-SSVM), which only requires the target domain samples together with the existing source-domain classifier for performing the desired adaptation. Altogether, we term our proposal as hierarchical A-SSVM (HA-SSVM).

As proof of concept we use HA-SSVM for pedestrian detection, object category recognition and face recognition. In the former we apply HA-SSVM to the deformable part-based model (DPM) while in the rest HA-SSVM is applied to multi-category classifiers. We will show how HA-SSVM is effective in increasing the detection/recognition accuracy with respect to adaptation strategies that ignore the structure of the target data. Since, the sub-domains of the target data are not always known a priori, we shown how HA-SSVM can incorporate sub-domain discovery for object category recognition.

J. Xu, S. Ramos, A. M. López
Computer Vision Center, Barcelona, Spain
Dept. of Computer Science, Universitat Autónoma de Barcelona, Barcelona, Spain
E-mail: {jiaolong, sramosp, antonio}@cvc.uab.es

D. Vázquez
Computer Vision Center, Barcelona, Spain
E-mail: dvazquez@cvc.uab.es

1 Introduction

Besides the data representations and learning algorithms used in classification tasks, other relevant fact that has been increasingly considered within this context is the so denominated *dataset bias*. This is a very common problem in real-world classification applications that makes the classifiers suffer from loss in accuracy when the data distribution in the training (*source*) domain differs from that in the testing (*target*) domain. In order to face this *domain adaptation* challenge a variety of methods have been increasingly explored within the machine learning field (Daumé III, 2007; Yang et al., 2007; Mansour et al., 2008; Ben-David et al., 2009; Duan et al., 2009; Pan and Yang, 2009) and most recently within the computer vision one (Bergamo and Torresani, 2010; Saenko et al., 2010; Gopalan et al., 2011; Duan et al., 2012; Vázquez et al., 2012, 2014; Hoffman et al., 2012, 2013; Kan et al., 2014) since image classification, visual object detection and object category recognition are tasks where dataset bias is usually relevant.

Many domain adaptation methods assume a single domain shift between the data, *i.e.*, they perform the adaptation from a single source domain to a single target domain (Daumé III, 2007; Bergamo and Torresani, 2010; Saenko et al., 2010; Duan et al., 2012; Vázquez et al., 2012, 2014; Hoffman et al., 2013; Kan et al., 2014). Some others consider multiple source domains (Yang et al., 2007; Mansour et al., 2008; Duan et al., 2009; Gopalan et al., 2011; Hoffman et al., 2012) and propose to leverage labeled data from them to perform the domain adaptation, *i.e.*, the underlying idea is to cover as much variability as possible at the source level for making more accurate predictions given a partially new domain (the target). In this paper we focus on the complementary case to these works. In other words, the main novelty is the study of the effectiveness of domain adaptation when we can structure the target domain as a hierarchy

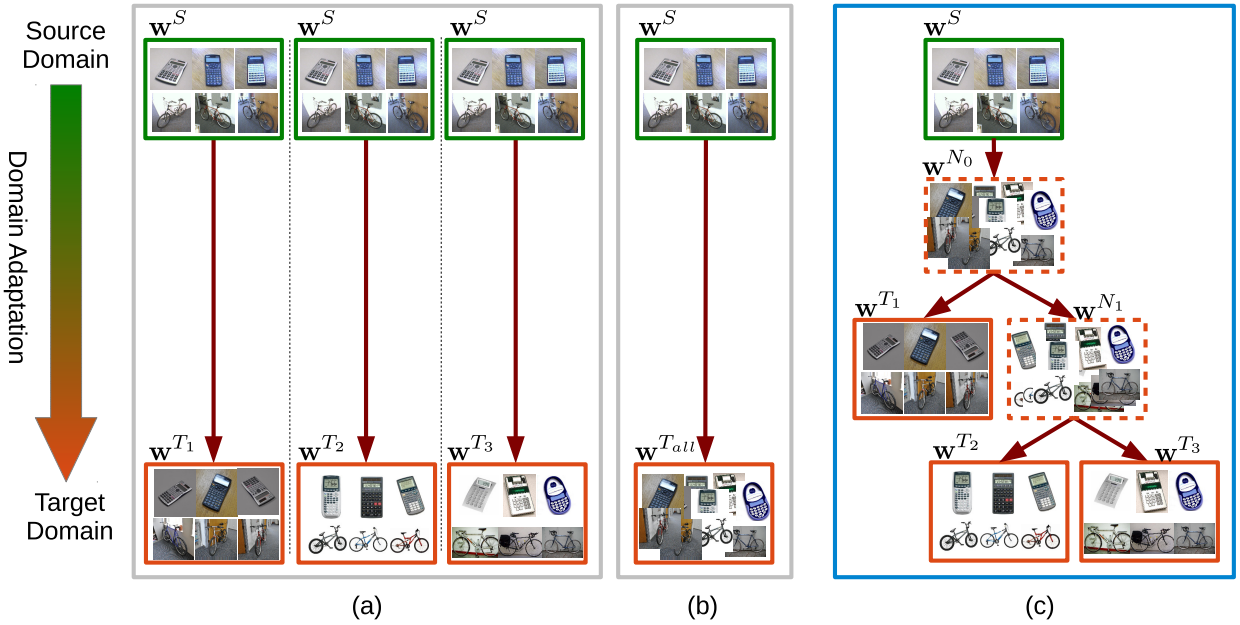

Fig. 1 Domain adaptation methods: without losing generality we assume a single source domain and three correlated target domains (three different datasets depicting the same object categories). (a) Single layer domain adaptation: adapting to each target domain $\mathbf{w}^{T_j}$ independently. (b) Single layer domain adaptation: pooling multiple target domains. (c) Proposed hierarchical multi-layer domain adaptation. The target domains are organized in an adaptation tree. Adaptation to intermediate nodes allows to exploit commonalities between children sub-domains, while adaptation to final sub-domains allows to consider their differences. Each path from the root to a leaf of the hierarchy can be thought as a progressive adaptation, but all models (intermediate and final) are learned jointly.

(*e.g.*, leveraging multiple correlated target domains or using some criteria to build sub-domain partitions). Moreover, due to its practical implications, we focus on methods that do not require to revisit the source data for the adaptation.

The main idea of our approach is illustrated in Fig. 1. Without losing generality assume that we have a prior source model $\mathbf{w}^S$ (*e.g.* a SVM hyperplane) and we would like to adapt it to multiple target domains (T_1, T_2, T_3) from which we have labeled data. Traditionally, $\mathbf{w}^S$ is adapted to each target domain separately, as illustrated in (a). Other option is to pool multiple target domains into a single one and adapt $\mathbf{w}^S$ to a mixed target domain as in (b). We refer to these strategies as single-layer domain adaptation.

Instead of performing isolated single-layer adaptations, we propose to make use of the relatedness of the target domains while exploiting their differences. Concretely, as it is presented in (c), we organize multiple target domains into a hierarchical structure (tree) and adapt the source model to them jointly. The adaptation to intermediate nodes allows to exploit commonalities between children sub-domains (*e.g.*, approach (b) is considered thanks to the root node of the hierarchy), while the adaptation to the final sub-domains allows to consider their differences.

Each path from the root to a leaf of the hierarchy can be thought as a progressive adaptation. However, as we will see, the adaptation of the whole hierarchy is done at once under the same objective function. This implies that our adaptation strategy is also useful in cases where the labeled data from the target domain is scarce but at the same time presents certain variability (sub-domains) worth to consider. Note that by using the approach in (a) such a reduced target domain dataset would be divided into even smaller target sub-domain datasets, which in general would end up in a poorer adaptation. On the other hand, following (b) the potential target sub-domains would be just ignored.

Given the widely use of SVM classifiers, we focus on the model-transform-based domain adaptation method known as *adaptive SVM* (A-SVM) (Yang et al., 2007). A-SVM does not require source domain samples, only target domain ones, which can significantly reduce the training (adaptation) time. In fact, since we will address problems requiring *structural SVM* (SSVM), we will use the A-SVM variation for SSVM that we introduced in (Xu et al., 2014a), namely the *adaptive SSVM* (A-SSVM). Altogether we term our approach as *hierarchical A-SSVM* (HA-SSVM).

As proof of concept we apply our method in pedestrian detection, object category recognition and face recognition. The former requires to use HA-SSVM with the widespread deformable part-based model (DPM) while the rest require to use HA-SSVM with multi-category classifiers. We will show how HA-SSVM is effective in increasing the classification accuracy with respect to state-of-the-art strategies that ignore the structure of the target data. Moreover, focusing on the object category recognition application, we will

also evaluate HA-SSVM in an scenario were the target sub-domains are not available a priori and must be discovered.

The rest of the paper is organized as follows. Section 2 overviews the related work. In section 3 we detail the general formulation of the proposed approach and its optimization method, as well as how to incorporate domain reshaping for discovering latent domains. Sections 4, 5 and 6 present the experimental results of HA-SSVM for pedestrian detection, object recognition and face recognition, respectively. Finally, section 7 draws the conclusions and future research lines.

2 Related Work

Despite the variety of domain adaptation methods proposed in the last decades–see (Jiang, 2008) for a comprehensive overview–, in computer vision, current methods can be broadly categorized in two main groups, namely feature-transform-based methods and model-transform-based methods.

Feature-transformation-based methods attempt to learn a transformation matrix/kernel over the feature space of different domains, and then apply a classifier (Saenko et al., 2010; Kulis et al., 2011; Gopalan et al., 2011; Gong et al., 2013b; Hoffman et al., 2014; Gong et al., 2014). For instance, the max-margin domain transforms method (Hoffman et al., 2013) jointly learns a feature transformation and a discriminative classifier via multi-task learning. Feature subspaces are generally used to connect the source and target domain, *e.g.*, (Gopalan et al., 2011) proposed to sample along the geodesic path using a few intermediate subspaces and (Gong et al., 2013b) further integrated an infinite number of intermediate subspaces. As such subspaces obtained using principle component analysis (PCA) may lose usefully information, recent works use dictionary to represent each domain and proposed domain-adaptive dictionary learning methods (Lu et al., 2015; Xu et al., 2015).

On the other hand, model-transform-based approaches concentrate on adapting the parameters of the classifiers, often SVM, including: weighted combination of source and target SVMs, transductive SVM (Bergamo and Torresani, 2010; Vázquez et al., 2012), feature replication (Daumé III, 2007; Vázquez et al., 2014), and regularization-based methods as A-SVM (Yang et al., 2007), its successor the projective model transfer SVM (PMT-SVM) (Aytar and Zisserman, 2011) and its variant A-SSVM (Xu et al., 2014a).

Among these methods, the SVM regularization-based ones have a significant advantage as they do not require revisiting source domain data for the adaptation. This would be favorable for many domain adaptation tasks in computer vision, since the source datasets are typically large and computing the features is expensive. Besides, it can even handle the case where the source data is missing at the moment of the adaptation. Basically, these methods learn the target classifier $f^T(\mathbf{x})$ by adding a perturbation function $\Delta f(\mathbf{x})$ to the source classifier $f^S(\mathbf{x})$ so that $f^T(\mathbf{x}) = f^S(\mathbf{x}) + \Delta f(\mathbf{x})$. Our approach can be regarded as a higher level model of A-SSVM, as it considers the hierarchical relationship between different domains, integrating multiple A-SSVM adaptations in a hierarchical model.

In the context of domain adaptation between multiple domains, several methods close to our work have been proposed in the natural language processing (NLP) community (Finkel and Christopher, 2009; Daumé III, 2009), which are Bayesian-based approaches (hierarchical Bayesian models and EM algorithm for inference). While for multi-domain adaptation most of the focus is on multiple sources, little attention is paid on the relationship of multiple target domains. Our domain adaptation method aims to leverage multiple target domains by considering their hierarchical relationship.

Most of the domain adaptation algorithms are validated assuming that the underlying domains are well-defined. However, multiple unknown domains may exist (Hoffman et al., 2012). In fact, in some cases image data is difficult to manually divide into discrete domains required by adaptation algorithms (Gong et al., 2013b). In (Hoffman et al., 2012), a sub-domain discovery algorithms is proposed, it focuses on discovering multiple hidden source domains. The most recent work of (Gong et al., 2013b) can discover domains among both training and testing data, which benefits existing multi-domain adaptation algorithms. In this paper, we also include experiments where HA-SSVM is applied to discovered target sub-domains for object category recognition.

One important feature of proposed hierarchical domain adaptation is the use of progressive adaptation strategy, *i.e.*, there are intermediate domains between source and target domain. Analogue ideas can be found in some recent literatures, *e.g.*, (Gong et al., 2013a) proposed to select landmarks, which are a subset of source domain data instances which are distributed most similarly to the target domain, and use them to construct easier auxiliary DA tasks. Each task represents different similarity level and thus provides progressively easier adaptation. (Tang et al., 2012) employed a self-paced learning strategy to gradually include more difficult examples from target domain and adapt the source SVM classifier towards the target domain. (Lu et al., 2015) proposed incremental dictionary learning which iteratively finds supportive samples from target domain and add them to source domain. The domain-adaptive dictionary learning of (Xu et al., 2015) generates a set of intermediate domains which bridge the gap between source and target domains. Nguyen et al. (2015) proposed a joint hierarchical feature learning method for domain adaptation, which learns common dictionary in multiple layers. The main difference between Nguyen et al. (2015) and our hierarchical domain

adaptation is that Nguyen et al. (2015) learns hierarchical features while our method learns hierarchical models.

The vision tasks where our method can be applied are several, however, in this paper our experiments focus on cross-domain multi-category object recognition and pedestrian detection based on the deformable part-based model (DPM). The former has been a benchmark for the proof of most domain adaptation methods developed for vision tasks (Saenko et al., 2010; Kulis et al., 2011; Gong et al., 2012). Despite its relevance, the latter has been just rarely addressed in the literature (Vázquez et al., 2012, 2014; Xu et al., 2014a) from the viewpoint of domain adaptation. However, it introduces the interesting challenges of dealing with rather unbalanced classes (*i.e.*, pedestrians *vs* background). Face recognition experiments are also included by considering view angle and illumination source as producing different domains with respect to a *neutral* pose and light.

Finally, it is worth to mention that the domain adaptation methods can be also divided as *supervised*, *unsupervised* and *semi-supervised*. In the supervised case it is assumed that target-domain adaptation data is annotated, in the unsupervised case it is assumed that target-domain data comes without annotations, and in the semi-supervised case both types of data are available (with and without annotations). It is assumed that in the supervised and semi-supervised cases the annotated target-domain data is relatively few, while in both the unsupervised and semi-supervised cases we can assume a relatively large amount of non-annotated target-domain data. HA-SSVM is a supervised domain adaptation method.

DASH-N: Joint Hierarchical Domain Adaptation and Feature Learning Nguyen et al. (2015)

Incremental Dictionary Learning for Unsupervised Domain Adaptation, (Lu et al., 2015)

Bridging the Domain Shift by Domain Adaptive Dictionary Learning (Xu et al., 2015)

Progressive DA (Gong et al., 2013a)

Shifting Weights: Adapting Object Detectors from Image to Video (Tang et al., 2012)

3 Proposed Method

3.1 General Model

Our proposal is illustrated in Fig. 1. Assume we have a prior model $\mathbf{w}^S$ from the source domain $\mathscr{D}^S$ and multiple target domains $\mathscr{D}^{T_j}, j \in [1,D]$. Traditionally, $\mathbf{w}^S$ is adapted to each target domain independently, as illustrated in (a), or to the pooled target domain as in (b), which we call single-layer domain adaptation in this paper. In contrast, we propose to make use of the relatedness of multiple target domains by combining them into a hierarchical adaptation tree, and adapt the prior model to them hierarchically, as in (c).

The proposed hierarchical model can be applied to any supervised learning algorithm which can incorporate prior information. In this work, we focus on the widely used SVM. This learning method considers a loss term $\mathscr{L}(\mathbf{w};\mathscr{D})$ that captures the error with respect to the training data $\mathscr{D}$ and a regularization term $\mathscr{R}(\mathbf{w})$ that penalizes model complexity. In fact, we will focus on domain adaptation with structural SVM (SSVM), giving rise to our hierarchical A-SSVM (HA-SSVM) in Sect 3.2.

3.2 Domain Adaptation Methods

For the sake of a better understanding, in this subsection, we introduce the involved concepts by progressive order of complexity. In Sect. 3.2.1 we focus on single-layer domain adaptation based on adaptive SVM (A-SVM). Then, in Sect. 3.2.2 we develop our hierarchical A-SVM (HA-SVM) model. We show how to learn its parameters by using a multiple task learning (MTL) paradigm. Finally, in Sect. 3.2.3 we consider SSVM and, therefore, introduce HA-SSVM.

3.2.1 Adaptive SVM (A-SVM)

A-SVM is a model-transform-based method, which adapts the model parameters from the source $\mathscr{D}_l^S$ to the target domain $\mathscr{D}_l^T$ (l indicates that the samples are labeled). Given the source domain model $\mathbf{w}^S$, the target domain model $\mathbf{w}^T$ is learned by minimizing the following objective function:

$$\min_{\mathbf{w}^T} \frac{1}{2}\|\mathbf{w}^T - \mathbf{w}^S\|^2 + C\mathscr{L}(\mathbf{w}^T;\mathscr{D}_l^T), \tag{1}$$

where the regularization term $\|\Delta\mathbf{w}\|^2 = \|\mathbf{w}^T - \mathbf{w}^S\|^2$ constrains the target model $\mathbf{w}^T$ to be close to the source one $\mathbf{w}^S$. Eq. (1) is also called one-to-one domain adaptation. At the testing time, we apply the following decision function to the target domain:

$$f^T(\mathbf{x}) = \mathbf{w}^{T\prime}\Phi(\mathbf{x}) = f^S(\mathbf{x}) + \Delta\mathbf{w}'\Phi(\mathbf{x}), \tag{2}$$

where $\Phi(\mathbf{x})$ is the feature vector for target domain sample $\mathbf{x}$ and $f^S(\mathbf{x})$ is the output score from the source domain classifier. Thus, A-SVM is essentially learning a perturbation function $\Delta f(\mathbf{x}) = \Delta\mathbf{w}'\Phi(\mathbf{x})$ based on the source classifier.

3.2.2 Hierarchical Adaptive SVM (HA-SVM)

We first state the general form of HA-SSVM for an arbitrary hierarchy of target domains. We assume that the depth of the target domain hierarchy is L. In the ith target-domain layer, $i \in \{1,\ldots,L\}$, there are M_i nodes (*i.e.* domains) enumerated from left to right as $\{1,\ldots,M_i\}$. Thus, (i,j) uniquely indexes the target-domain of layer i at position $j \in \{1,\ldots,M_i\}$. We assume $M_1 = 1$, *i.e.* there is a single target-domain acting

as root of the hierarchy of target sub-domains; the source domain is connected to it. The objective function of HA-SVM can be written as:

$$J(\mathbf{w}) = \sum_{i=1}^{L}\sum_{j=1}^{M_i}[\frac{1}{2}\|\mathbf{w}^{i,j}-\mathbf{w}^{p(i,j)}\|^2 + C\mathscr{L}(\mathbf{w}^{i,j};\mathscr{D}_l^{i,j})], \tag{3}$$

where $\mathbf{w}^{i,j}$ stands for the parameters at target-domain (i,j), $\mathbf{w}$ is the concatenation of all such parameters, $p(i,j)$ indexes the parent domain of (i,j), where $p(1,1)$ is the source S, and $\mathscr{D}_l^{i,j}$ contains the training samples of the target domain (i,j), *i.e.* the union of the training samples in the child target sub-domains of (i,j).

At testing time, for a given sample $\mathbf{x}$ that we consider as belonging to target domain (i,j), we can directly consider the learned parameters $\mathbf{w}^{(i,j)}$ and apply the linear decision function:

$$f^{i,j}(\mathbf{x}) = \mathbf{w}^{i,j'}\boldsymbol{\Phi}(\mathbf{x})\ . \tag{4}$$

The domain (i,j) can be a leaf of the hierarchy or an intermediate node, depending on the capability to discern the target domain of the given testing sample. In the worst case the root of the hierarchy can be used, *i.e.* $\mathbf{w}^{1,1}$. Note that, in this way, the hierarchy allows to classify a given sample with the model that is the most specific for the domain information that it has been possible to infer for the sample at testing time. For instance, imagine that we must detect an object in a sequence with a new camera sensor (different from the one used to train the source model) and we have organized the target domain by aspect. Then, this aspect can be tracked with an associated uncertainty. Therefore, we can apply a more specific sub-domain model (leaf) if the uncertainty is low, and a more general sub-domain model (intermediate/root) otherwise. In general terms, if the criterion used to determine the target domain for a given sample comes with an uncertainty value, we could design an application-dependent strategy for classifying the sample using the model of one target sub-domain or another (root, leaf, intermediate), so that the same sample could be classified according to different models of the hierarchy just depending on the mentioned certainty value.

For the sake of simplicity but without losing generality, in the following we analyze the formulation of HA-SVM for the hierarchy of three layers illustrated in Fig. 1 (c), using the notation of this figure too (note that $(1,1)\leftrightarrow N_0$, $(2,1)\leftrightarrow T_1$, $(2,2)\leftrightarrow N_1$, $(3,1)\leftrightarrow T_2$, $(3,2)\leftrightarrow T_3$). Accordingly, the objective function of this three-layers HA-SVM can be written as follows:

$$\begin{aligned} J(\mathbf{w}) = &\frac{1}{2}\|\mathbf{w}^{N_0}-\mathbf{w}^{S}\|^2 + C\mathscr{L}(\mathbf{w}^{N_0};\cup_{j=1}^{3}\mathscr{D}_l^{T_j}) \\ &+\frac{1}{2}\|\mathbf{w}^{N_1}-\mathbf{w}^{N_0}\|^2 + C\mathscr{L}(\mathbf{w}^{N_1};\cup_{j=2}^{3}\mathscr{D}_l^{T_j}) \\ &+\frac{1}{2}\|\mathbf{w}^{T_1}-\mathbf{w}^{N_0}\|^2 + C\mathscr{L}(\mathbf{w}^{T_1};\mathscr{D}_l^{T_1}) \\ &+\frac{1}{2}\|\mathbf{w}^{T_2}-\mathbf{w}^{N_1}\|^2 + C\mathscr{L}(\mathbf{w}^{T_2};\mathscr{D}_l^{T_2}) \\ &+\frac{1}{2}\|\mathbf{w}^{T_3}-\mathbf{w}^{N_1}\|^2 + C\mathscr{L}(\mathbf{w}^{T_3};\mathscr{D}_l^{T_3}) \end{aligned} \tag{5}$$

Eq. (5) is in a multi-task learning paradigm form, where the optimization of each $\mathbf{w}^{T_j}$ can be understood as an individual task. The key issue of the multi-task learning lies in how the relationships between tasks are incorporated. As we can see from Eq. (5), each task is related by the regularization term, *e.g.*, T_2 and T_3 are connected by $\|\mathbf{w}^{T_j}-\mathbf{w}^{N_1}\|^2$, while T_1 is directly connected to N_0, which is adapted from $\mathbf{w}^S$.

Comparing to the single-layer adaptation $\|\mathbf{w}^{T_j}-\mathbf{w}^S\|^2$ as in Fig. 1 (a), HA-SVM has several advantages. First, HA-SVM can make use of training samples from multiple related target domains instead of just one. For example, a single-layer domain adaptation only uses the training samples from $T_j, j\in\{1,2,3\}$ in three different optimization runs, while HA-SVM can integrate the samples from the three target domains accounting for their hierarchical structure. Second, the target model $\mathbf{w}^{T_j}$ is not directly regularized by $\mathbf{w}^S$ but some shared intermediate models $\mathbf{w}^{N_i}$, which allows $\mathbf{w}^{T_j}$ to be regularized in a way less constrained by the source model. As $\mathbf{w}^{T_j}$ goes down apart from $\mathbf{w}^S$ further in the adaptation tree, less constrain from $\mathbf{w}^S$ is imposed. This can be interpreted as a progressive adaptation.

For single-layer domain adaptation, another straightforward strategy is to pool all target domains and train a single adaptive SVM with all available target samples, as illustrated in Fig. 1 (b). Comparing to this method, HA-SVM can take the same advantage of using all available labeled data while allows each target domain model to be more discriminative in its own domain. The pooling-based method requires the final model to compromise to each domain in order to minimize the training error, and thus such model may lose the discriminative power in the individual target domains. Our experimental results in Sect. 4.1 and Sect. 5.1 confirm this observation.

To minimize Eq. (3), we employ Quasi-Newton LBFGS method, which requires the objective function and the partial derivatives of its parameters. For instance, for our guiding

example of Eq. (5), these partial derivatives are:

$$\begin{aligned}
\frac{\partial J(\mathbf{w})}{\partial \mathbf{w}^{N_0}} &= 3\mathbf{w}^{N_0} - \mathbf{w}^{S} - \mathbf{w}^{N_1} - \mathbf{w}^{T_1} + C\frac{\partial \mathscr{L}(\mathbf{w}^{N_0}; \cup_{j=1}^{3} \mathscr{D}_l^{T_j})}{\partial \mathbf{w}^{N_0}} \ , \\
\frac{\partial J(\mathbf{w})}{\partial \mathbf{w}^{N_1}} &= 3\mathbf{w}^{N_1} - \mathbf{w}^{N_0} - \mathbf{w}^{T_1} - \mathbf{w}^{T_2} + C\frac{\partial \mathscr{L}(\mathbf{w}^{N_1}; \cup_{j=2}^{3} \mathscr{D}_l^{T_j})}{\partial \mathbf{w}^{N_1}} \ , \\
\frac{\partial J(\mathbf{w})}{\partial \mathbf{w}^{T_1}} &= \mathbf{w}^{T_1} - \mathbf{w}^{N_0} + C\frac{\partial \mathscr{L}(\mathbf{w}^{T_1}; \mathscr{D}_l^{T_1})}{\partial \mathbf{w}^{T_1}} \ , \\
\frac{\partial J(\mathbf{w})}{\partial \mathbf{w}^{T_2}} &= \mathbf{w}^{T_2} - \mathbf{w}^{N_1} + C\frac{\partial \mathscr{L}(\mathbf{w}^{T_2}; \mathscr{D}_l^{T_2})}{\partial \mathbf{w}^{T_2}} \ , \\
\frac{\partial J(\mathbf{w})}{\partial \mathbf{w}^{T_3}} &= \mathbf{w}^{T_3} - \mathbf{w}^{N_1} + C\frac{\partial \mathscr{L}(\mathbf{w}^{T_3}; \mathscr{D}_l^{T_3})}{\partial \mathbf{w}^{T_3}} \ .
\end{aligned} \tag{6}$$

In our implementation, the LBFGS based optimization converges to the optimum efficiently for both single-layer A-SVM and HA-SVM (as well as for the HA-SSVM defined in next subsection).

3.2.3 Hierarchical Adaptive Structured SVM (HA-SSVM)

The proposed HA-SVM can be extended for SSVM, giving rise to our HA-SSVM. SSVM generalized SVM and allows the training of a classifier for general structured output labels. In this work, we use SSVM as an unified classifier for both multiple category classification and part-based object detection. SSVM minimizes the following regularized risk function:

$$\begin{aligned}
&\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|^2 \\
&+C\textstyle\sum_{i=1}^{N}[\max_y \mathbf{w}'\boldsymbol{\Phi}(\mathbf{x}_i, y) + \Delta(y_i, y) - \boldsymbol{\Phi}(\mathbf{x}_i, y_i)] \ ,
\end{aligned} \tag{7}$$

where y_i is the ground truth output (label) of sample $\mathbf{x}_i$, and y runs on the alternative outputs. $\Delta(y_i, y)$ is a distance in output space. $\boldsymbol{\Phi}(\mathbf{x}, y)$ is the feature vector from a given sample $\mathbf{x}$ of label y. Accordingly, Eq. (1) can be extended to A-SSVM as follows:

$$\begin{aligned}
&\min_{\mathbf{w}^T} \frac{1}{2}\|\mathbf{w}^T - \mathbf{w}^S\|^2 \\
&+C\textstyle\sum_{i=1}^{N}[\max_y \mathbf{w}^{T\prime}\boldsymbol{\Phi}(\mathbf{x}_i, y) + \Delta(y_i, y) - \boldsymbol{\Phi}(\mathbf{x}_i, y_i)] \ .
\end{aligned} \tag{8}$$

Correspondingly, the final adapted classifier f^T can be written as:

$$f^T(\mathbf{x}) = \max_y [\mathbf{w}^{S\prime}\boldsymbol{\Phi}(\mathbf{x}, y) + \underbrace{\Delta\mathbf{w}'\boldsymbol{\Phi}(\mathbf{x}, y)}_{\Delta f(\mathbf{x})}] \ , \tag{9}$$

where $\Delta\mathbf{w} = \mathbf{w}^T - \mathbf{w}^S$. Therefore, Eq. (8) can be integrated into the proposed hierarchical adaptation framework described in Section 3.2.2. In particular, we must proceed in the same way than going from (1) to (3), but now starting with (8); thus, giving rise to HA-SSVM.

4 Domain Adaptation of DPMs

In this section we apply HA-SSVM in the popular deformable part-based model (DPM) framework (Felzenszwalb et al., 2010), focusing on pedestrian detection. The DPM learns a linear classification parameter $\mathbf{w}$, which parametrizes a set of parts and deformations, to decide whether a detection window contains a pedestrian or background. The learning of a DPM is usually formulated as the latent SVM (LSVM) framework (Felzenszwalb et al., 2010). However, it is also possible to use a latent structural SVM (LSSVM) formulation (Zhu et al., 2010; Girshick, 2012), which can be solved by the Convex-Concave Procedure (CCCP). The LSSVM form of a DPM objective function can be written as:

$$\begin{aligned}
\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|^2 \ &+C\textstyle\sum_{i=1}^{N}[\max_{\mathbf{h}}(\mathbf{w}'\boldsymbol{\Phi}(\mathbf{x}_i, \mathbf{h}) \\
&+\Delta(y, y_i)) - \mathbf{w}'\boldsymbol{\Phi}(\mathbf{x}_i, \mathbf{h}^*)],
\end{aligned} \tag{10}$$

where $\mathbf{h}$ is the latent variable which defines the object hypothesis, *e.g.*, the alignment of parts, $\boldsymbol{\Phi}(\mathbf{x}, \mathbf{h})$ concatenates HOG-based (Dalal and Triggs, 2005; Felzenszwalb et al., 2010) appearance features and part spatial alignment features, $\Delta(y, y_i)$ is the 0-1 loss function that returns 0 if the binary label y equals y_i, and 1 otherwise. $\mathbf{h}^*$ plays the role of ground truth output of example i and it is computed at the concave stage of CCCP. We refer to (Zhu et al., 2010) (Girshick, 2012) for more details. A-SSVM can be applied to DPM according to Eq. (8). For example, A-SSVM for DPM can be written as:

$$\begin{aligned}
\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}^T - \mathbf{w}^S\|^2 \ &+C\textstyle\sum_{i=1}^{N}[\max_{\mathbf{h}}(\mathbf{w}^{T\prime}\boldsymbol{\Phi}(\mathbf{x}_i, \mathbf{h}) \\
&+\Delta(y, y_i)) - \mathbf{w}^{T\prime}\boldsymbol{\Phi}(\mathbf{x}_i, \mathbf{h}^*)].
\end{aligned} \tag{11}$$

Thus, we can build HA-SSVM for DPM. We implemented it in the DPM 5.0 framework (Girshick et al., 2012), which is the latest at the moment of doing this research. When applying an adapted DPM in a particular target domain (*i.e.*, in testing time), we do not use the full vector of parameters jointly learned for all the hierarchy of target domains, instead we only use the sub-vector of parameters corresponding to such particular target domain (*i.e.* as mentioned for Eq. (4)).

4.1 Experiments on Pedestrian Detection

Fig. 2 illustrates two different cases of DPM domain adaptation using HA-SSVM that we evaluate here. In (a) the source classifier is adapted to three different target domains (different datasets in this case). In (b) we adapt the source classifier to detect pedestrians from image windows of two different resolution categories. The main idea is to divide the target domain into sub-domains according to the resolution of the

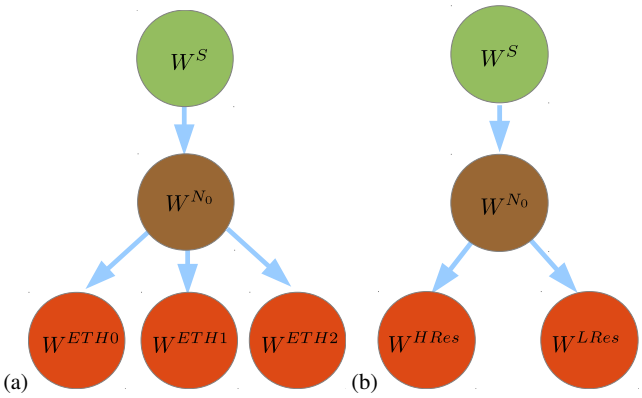

Fig. 2 HA-SSVM applied to DPM: (a) adaptation to three related datasets (domains), namely ETH0, ETH1 and ETH2, which were acquired with the same camera but different environments; (b) adaptation of a single resolution detector for applying different detectors when processing small and large image windows, which would correspond to pedestrians imaged with low (LRes) and high (HRes) resolution respectively.

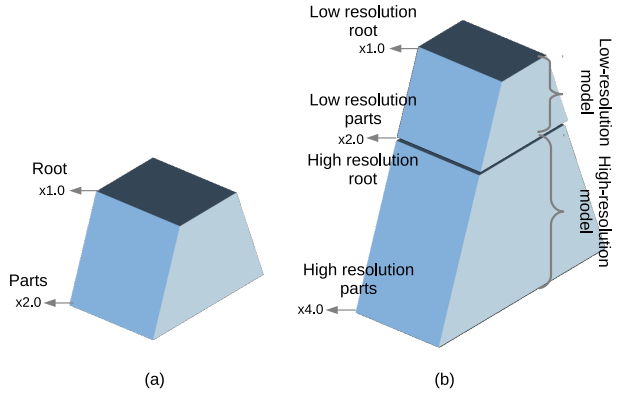

Fig. 3 (a) Original feature pyramid in DPM. (b) The extended feature pyramid for multi-resolution adaptive DPM.

pedestrian samples, *i.e.*, different resolutions are regarded as different domains. Here we consider only two resolutions, low and high. Note that low resolution pedestrians tend to be blur and their poses are less discriminative than for high resolution ones.

4.1.1 Datasets

As source-domain virtual-world dataset[1] (Fig. 4) we use the same as in (Vázquez et al., 2014; Xu et al., 2014a,b). The target-domain real-world datasets that we use are ETH (Ess et al., 2007), Caltech (Dollár et al., 2012) and KITTI (Geiger et al., 2012). The ETH dataset consists of three sub-datasets, namely ETH0 (BAHNHOF), ETH1 (JELMOLI) and ETH2 (SUNNY DAY), and is used to evaluate the setting of Fig. 2(a). These sub-datasets are collected from different environments

[1] It is publicly available under the name *CVC-07 DPM Virtual-World Pedestrian Dataset* at `www.cvc.uab.es/adas`

Fig. 4 Virtual-world pedestrians and background images.

SRC	Classifier trained with only the source (virtual-world) domain samples.
TAR	Classifier trained with only a relatively low number of target domain (real-world) samples.
MIX	Classifier trained with source samples used for SRC and the target domain samples used for TAR.
A-SSVM	Classifier adapted with A-SSVM using the SRC model and the target domain samples used for TAR.
A-SSVM-ALL	As before but pooling all the considered target domains.

Table 1 Different types of learned DPM classifiers.

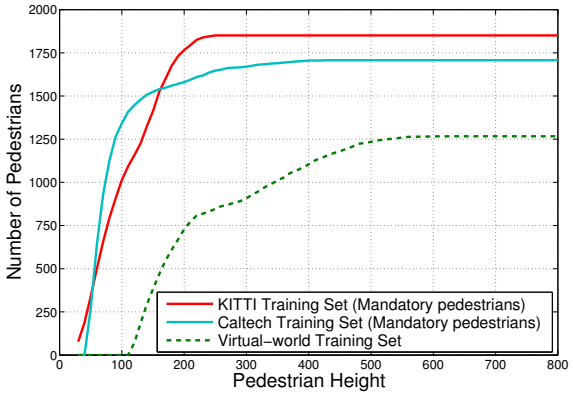

Fig. 5 Cumulative histogram of the pedestrians' height in Caltech, KITTI and virtual-world training dataset. The virtual-world dataset contains less low-resolution pedestrians than the real-world ones.

but with the same camera sensor and pose, thus, we consider them as related sub-domains. Caltech and KITTI are used in two different experiments, both for evaluating the setting of Fig. 2(b).

4.1.2 Setup

In supervised domain adaptation it is assumed that there are available just a few labeled data from the different target domains. In order to emulate this setting, we selected only 100 pedestrians for the experiments with ETH0, ETH1, ETH2 and Caltech, which roughly correspond to the 6%,

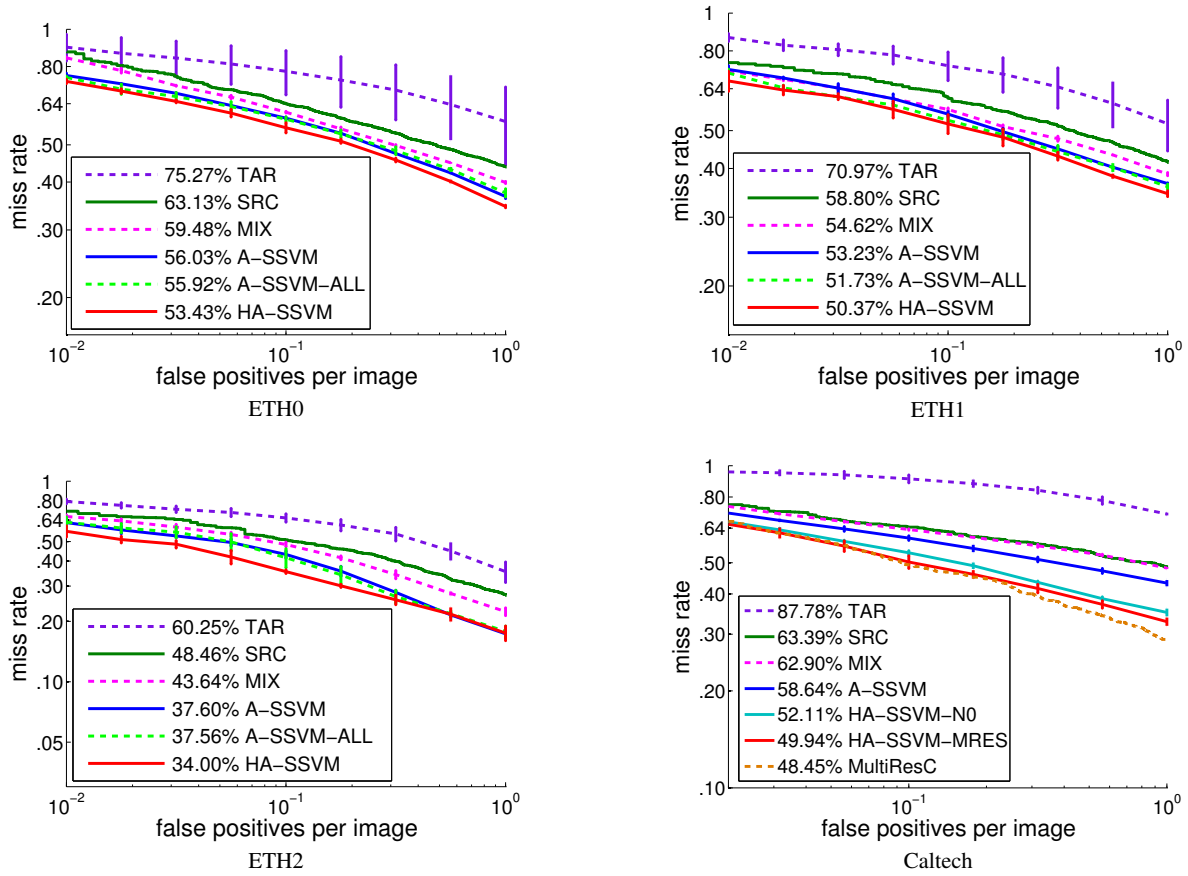

Fig. 6 Average miss rate among the FPPI range from 0.01 to 1. ETH0, ETH1 and ETH2 show the adaptation results from virtual-world to ETH three sub-datasets. Caltech shows results of adapting virtual-world DPM detector to a multi-resolution detector in Caltech pedestrian dataset. A-SSVM is trained with mixed high and low resolution samples. HA-SSVM-N0 corresponds to W^{N_0} in the multi-resolution adaptation tree and HA-SSVM-MRES is the adapted multi-resolution detector. MultiResC is the multi-resolution DPM proposed in (Park et al., 2010).

1.5%, 3%, 5%, respectively, of the available pedestrians for training. We use all the training pedestrian-free images of these datasets, *i.e.*, 999, 451, 354, 1, 824 images, respectively. We follow the Caltech evaluation criterion (Dollár et al., 2012) and plot the average miss rate *vs* false positive per image (FPPI) curve. We use the suggested reasonable setting and therefore test on the pedestrians taller than 50 pixels. Each train-test experiment is repeated five times and we report the mean and standard deviation of the repetitions. To ensure fair comparisons, we use the same random samples for different training methods. To evaluate the performance of HA-SSVM, we compare it to the baselines described in Table 1.

In fact, as touchstone of HA-SSVM for pedestrian detection, our first test was the participation in the *Pedestrian Detection Challenge* of the KITTI benchmark[2] as part of the *Reconstruction Meets Recognition Challenge (RMRC)* held in conjunction with the ICCV'2013 celebrated in Sydney. At that time we did not have written neither a report nor this paper, so we participated with the multi-resolution HA-SSVM DPM described here, but with the generic name of *DA-DPM* (domain adaptive DPM). In this case, we used 200 pedestrians of the KITTI training set, roughly the 11% of the available ones, as well as 2,000 pedestrian-free images of the 7,518 available for training. We note that under the KITTI benchmark, the object detection evaluation criterion is different from the Caltech one. Accuracy is measured as precision *vs* recall instead of miss rate *vs* FPPI. Note also that, in order to avoid parameter tuning, the ground truth of the KITTI testing data is not available.

For all the experiments, the SRC classifier (see Table 1) is the same DPM, trained with the virtual-world dataset. It is worth to mention that we use a DPM root filter of 12×6 HOG cells (each cell is of 8×8 pixels), *i.e.*, the minimum size of the detectable pedestrians is 96×48 pixels. Then, for the multi-resolution adaptation (to Caltech and KITTI), we build the two-layer hierarchical model of Fig. 2(b). When computing features, we add an extra octave at the bottom of the feature pyramid and then divide the pyramid into two pyramids: high resolution pyramid which contains pedestri-

[2] www.cvlibs.net/datasets/kitti/

Rank	Method	Moderate	Easy	Hard
1	HA-SSVM	**45.51 %**	**56.36 %**	**41.08 %**
2	LSVM-MDPM-sv	39.36 %	47.74 %	35.95 %
3	LSVM-MDPM-us	38.35 %	45.50 %	34.78 %
4	mBoW (LP)	31.37 %	44.28 %	30.62 %

Table 2 Evaluation on KITTI pedestrian detection benchmark during the RMRC'2013. Results are given as average precision (AP). Our method HA-SSVM (which is actually the application of HA-SSVM to a DPM trained with virtual-world data) outperforms the previous best method LSVM-MDPM-sv by $5 \sim 8$ points in average precision. "LP" means that the method uses point clouds from a Velodyne laser scanner.

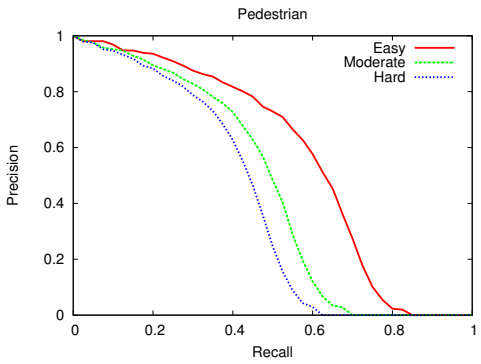

Fig. 8 Pedestrian detection results on KITTI benchmark.

ans taller than 96 pixels, and low resolution pyramid which contains pedestrians lower than 96 pixels. The extended feature pyramid is illustrated in Fig. 3(b). During training time, we assign the training pedestrians to the high and low resolution domains according to the height of their bounding boxes, while the background samples are shared by both domains. In figure Fig. 5 we show the pedestrian height distribution of the virtual- and real-world training datasets. Note that the virtual-world dataset has few low resolution pedestrians compared with the real-world ones, thus making the pursued adaptation challenging. In testing time, we apply the two adapted models to the corresponding resolution pyramid and finally we combine their detections and apply non-maximum suppression to obtain the final detections.

Note that in the particular case of using virtual-data we have the chance of generating the desired number of pedestrians at any given resolution. However, with this experiment we want to illustrate how HA-SSVM can be used for simultaneously adapting to a new sensor and new predominant resolutions, irrespectively of the origin of the source data. On the other hand, in the general case, it is not always possible improving the source model in terms of the resolution discrepancy before sensor adaptation. The reason is that the user of the model may not be the owner of the source data used to train it. Moreover, in this case seems to be worth to expend our resources in annotating the target-domain data rather than collecting and annotating source-domain data.

4.1.3 Results

In Fig. 6 we can see the results for the setting of Fig. 2(a). It can be appreciated that pooling-all-target-domains strategy (A-SSVM-ALL) can have better adaptation accuracy than using single target domain data (A-SSVM). However, HA-SSVM achieves even better accuracy when trained with the same samples as A-SSVM-ALL, which demonstrates the importance of leveraging multiple target domains in a hierarchy.

In Fig. 6 we can also see the results of applying setting Fig. 2(a) to Caltech. We additionally assessed the accuracy provided by the intermediate model $\mathbf{w}^{N0}$, which is denoted by HA-SSVM-N0 in contrast to HA-SSVM-MRes which corresponds to the full multi-resolution adaptation. Note how even HA-SSVM-N0 shows better classification accuracy than A-SSVM. It demonstrates the effectiveness of the progressive adaptation in HA-SSVM, which can be explained by the fact that the multi-task training learns general shared parameters for multiple target domains (*i.e.* high- and low-resolution domains), while single-task A-SSVM does not take into account the differences of multi-resolution samples. Of course, HA-SSVM-MRes is providing the best accuracy. Quantitative results are shown in Fig. 7, where it can be seen that HA-SSVM-MRes is capable of detecting lower resolution pedestrians.

Finally, as can be see in Table 2, we won the pedestrian detection challenge of the RMRC'2013[3], *i.e.*, we outperformed LSVM-MDPM-sv (A.Geiger et al., 2011), LSVM-MDPM-us (Felzenszwalb et al., 2010) and mBoW (Behley et al., 2013). Our precision-recall curves can be seen in Fig. 8.

We think that this was a quite remarkable result because, as we mentioned before, in order to adapt our virtual-world based pedestrian DPM we only used the $\sim 11\%$ of the KITTI training pedestrians and the $\sim 27\%$ of the available pedestrian-free images. This implies that the adaptation took 20 minutes approximately in our 1 core @ 3.5 Ghz desktop computer, while training the original DPM (Felzenszwalb et al., 2010) with all the full KITTI training set may need around 10 hours in the same conditions. It is also worth to point out that, as can be deduced from Fig. 5, the number of (virtual-world) pedestrians used for building our source model plus the 200 pedestrians selected from the KITTI training dataset is still lower than the total number of pedestrians in the KITTI dataset. Similarly for the pedestrian-free images, since we use $2,000$ from the virtual world to build the source model and $2,000$ from the KITTI training set to do the adaptation, while there are around $7,500$ available for training. Moreover, as to the best of our knowledge, this

[3] In the RMRC'2013 program it can be checked that we did a talk as winners of the pedestrian detection challenge, see `http://ttic.uchicago.edu/~rurtasun/rmrc/program.php`

Fig. 7 Pedestrian detection on Caltech dataset for SRC, A-SSVM and HA-SSVM based models. The results are drawn at FPPI = 0.1. HA-SSVM improves SRC and A-SSVM not only by low resolution detection but also by other factors, *e.g.*, reducing false positives (column 2) and better localization accuracy (column 4).

was the first time that a domain adapted object detector wins such a challenge.

Of course, after RMRC'2013 we were completing the work presented in this paper for the initial and following submission, many other proposals have been submitted to the pedestrian detection challenge[4], some of them complementing RGB information with LIDAR or Stereo data, or using Deep Learning techniques, other are just anonymous. As a result our HA-SSVM is now roughly in the middle of the classification. However, it is still the first method based on the DPM applied to only RGB data, even performing better than submissions done after ours based on DPM, see DPM-VOC+VP (Pepikj et al., 2015) which improves the DPM considering 3D view pose, and DPM-C8B1 (Yebes et al., 2015) which even uses 3D stereo information to complement RGB data. Among DPM methods it is just slightly outperformed by Fusion-DPM (Premebida et al., 2014) which makes use of LIDAR and RGB data, and it slower too. We plan to improve our detector in terms of used features and pedestrian model; however, this is out of the scope of this paper since these improvements are not related to the domain adaptation process itself. In spited of this, we tried to improve our result just by using all the available training data of KITTI for the domain adaptation. However, the result was not significantly better. We think that is because the *standard* DPM cannot learn more from this data. In fact, as we have mentioned, there is just one DPM based method above ours and it is because the use of an additional data modality (LIDAR).

[4] `www.cvlibs.net/datasets/kitti/eval_object.php`

5 Domain Adaptation of Multi-category Classifiers

In the following, we evaluate HA-SSVM on multi-category classifiers. We begin with the scenario in which the target sub-domains are given a priori. After we assess the scenario in which such sub-domains must be discovered. For illustrating how HA-SSVM operates with multi-category classifiers, we focus on object category recognition.

Assume we are given an set of N examples, $\mathscr{D}$, each one labeled as belonging to a category among K possible ones, *i.e.* $\mathscr{D} = \{(\mathbf{x}_i, y_i) | \mathbf{x}_i \in \mathbb{R}^n, y_i \in \{1,\dots,K\}\}_{i=1}^N$. Let $\mathbf{w}_1,\dots,\mathbf{w}_K$ be the parameters of K linear category classifiers, so that a new example $\mathbf{x}$ is assigned to a category according to the rule $f(\mathbf{x}) = \text{argmax}_{k\in\{1,\dots,K\}} \mathbf{w}_k'\mathbf{x}$. Let $\mathbf{w} = [\mathbf{w}_1',\dots,\mathbf{w}_K']'$ in $\mathbb{R}^{Kn}$, and a feature map $\Phi(\mathbf{x},y) = [\mathbf{0}',\dots,\mathbf{x}',\dots,\mathbf{0}']'$, where $\mathbf{0} \in \mathbb{R}^n$ is the zero vector and $\mathbf{x}$ is located at the y-th slot in $\Phi(\mathbf{x},y)$. Now the multi-category classification problem can be treated as a special case of structure output prediction:

$$f(\mathbf{x}) = \underset{y\in\{1,\dots,K\}}{\text{argmax}}\ \mathbf{w}'\Phi(\mathbf{x},y). \tag{12}$$

In order to apply HA-SSVM, Eq. (8) can be directly used as a basic adaptation unit by writing the loss term as:

$$\sum_{i=1}^N [\max_{\hat{y}\in\{1,\dots,K\}} (\mathbf{w}'\Phi(\mathbf{x},\hat{y}) + \Delta(\hat{y},y_i)) - \mathbf{w}'\Phi(\mathbf{x}_i,y_i)], \tag{13}$$

where $\Delta(\hat{y},y_i)$ is the 0-1 loss function.

5.1 Known Target Sub-Domains

	A → W	A → D	A → C	W → A	W → D	W → C
ASVM(Yang et al., 2007)	65.0 ± 1.0	51.6 ± 1.1	30.9 ± 0.6	48.6 ± 1.1	54.4 ± 1.5	29.8 ± 1.0
PMT-SVM(Aytar and Zisserman, 2011)	65.9 ± 1.0	52.6 ± 1.1	32.3 ± 0.6	49.0 ± 1.1	57.9 ± 1.6	30.4 ± 0.9
GFK(Gong et al., 2012)	56.5 ± 0.8	45.3 ± 0.9	38.6 ± 0.4	45.8 ± 0.6	**73.8** ± 0.7	32.6 ± 0.6
MMDT(Hoffman et al., 2013)	65.1 ± 1.2	54.5 ± 1.0	39.7 ± 0.5	50.6 ± 0.8	62.5 ± 1.0	34.8 ± 0.8
A-SSVM	60.0 ± 0.9	49.7 ± 0.8	**42.6** ± 0.5	49.5 ± 0.5	67.4 ± 0.7	37.3 ± 0.5
A-SSVM-ALL	64.5 ± 0.7	55.1 ± 0.8	**42.6** ± 0.3	49.8 ± 0.5	67.8 ± 0.8	39.0 ± 0.3
HA-SSVM	**69.8** ± 0.7	**59.7** ± 0.9	42.1 ± 0.4	**54.4** ± 0.6	66.1 ± 1.1	**39.4** ± 0.3

	D → A	D → W	D → C	C → A	C → W	C → D
ASVM(Yang et al., 2007)	48.0 ± 1.1	63.5 ± 1.1	29.9 ± 0.8	49.5 ± 1.0	63.2 ± 1.2	52.7 ± 1.3
PMT-SVM(Aytar and Zisserman, 2011)	48.6 ± 1.1	66.5 ± 1.2	30.9 ± 0.8	50.0 ± 1.0	64.3 ± 1.2	52.2 ± 1.3
GFK(Gong et al., 2012)	45.8 ± 0.4	**80.3** ± 0.7	33.3 ± 0.5	46.4 ± 0.7	61.0 ± 1.4	52.7 ± 1.2
MMDT(Hoffman et al., 2013)	50.4 ± 0.7	74.2 ± 0.7	35.7 ± 0.7	51.1 ± 0.7	62.9 ± 1.1	53.0 ± 1.0
A-SSVM	48.6 ± 0.5	74.6 ± 0.6	35.5 ± 0.5	**53.4** ± 0.7	63.6 ± 1.2	52.7 ± 1.0
A-SSVM-ALL	49.0 ± 0.5	73.2 ± 0.7	37.8 ± 0.4	50.6 ± 0.6	66.2 ± 0.6	57.5 ± 0.9
HA-SSVM	**52.6** ± 0.5	73.0 ± 0.5	**39.2** ± 0.6	**53.4** ± 0.8	**69.6** ± 0.7	**61.2** ± 0.9

Table 4 Multi-category recognition accuracy on target domains. Bold indicates the best result for each domain split. Underline indicates the second best result. The domains are: A: *amazon*, W: *webcam*, D: *dslr*, C: *Caltech256*. We use the nomenclature *Source→Target*.

ASVM	PMT-SVM	GFK	MMDT	A-SSVM	A-SSVM-ALL	HA-SSVM
48.9 ± 1.1	48.9 ± 1.1	51.0 ± 0.7	52.9 ± 0.9	52.9 ± 0.7	54.4 ± 0.6	**56.7** ± 0.7

Table 5 Multi-category recognition accuracy averaged across all domain splits, with the corresponding standard deviation.

ASVM	Adaptive SVM (Yang et al., 2007). It does not require the source domain data, only the learned source classifier. In contrast to A-SSVM, ASVM does not consider structural information.
PMT-SVM	Projective model transfer SVM (Aytar and Zisserman, 2011), which is a variant of ASVM.
GFK	The geodesic flow kernel method (Gong et al., 2012), which requires both source and target domain data (including testing data). Labels of target training data are not required.
MMDT	Max-margin domain transfer method of (Hoffman et al., 2013), which learns a mapping from target domain to source domain as well as a discriminative classifier using the mapped target and source domain features.
A-SSVM	Analogous to Table 1.
A-SSVM-ALL	Analogous to Table 1.

Table 3 Different types of learned multi-category classifiers.

5.1.1 Datasets

We evaluate HA-SSVM for object category recognition using the benchmark domain adaptation dataset known as *Office-Caltech* (Gong et al., 2012; Hoffman et al., 2013). This dataset combines the *Office* (Saenko et al., 2010) and *Caltech256* (Griffin et al., 2007) datasets. In particular, *Office-Caltech* consists of the 10 overlapping object categories between *Office* and *Caltech256*, which are `backpack`, `calculator`, `coffee-mug`, `computer-keyboard`, `computer-monitor`, `computer-mouse`, `head-phones`, `laptop-101`, `touring-bike` and `video-projector` in the terminology of *Caltech256*.

From the viewpoint of domain adaptation, *Office-Caltech* consists of four domains. One domain, called *caltech* (C), corresponds to the images of *Caltech256*, which were collected from the internet using Google. The other three domains come from *Office*, namely the *amazon* (A), *webcam* (W) and *dslr* (D) domains. The *amazon* domain is a collection of product images from `amazon.com`. The *webcam* and *dslr* domains contain images taken by a (low resolution) webcam and a (high resolution) digital single-lens reflex camera, respectively. Cross-domain variations are not the only ones, but for a particular domain and category, the objects are imaged under different poses and illumination conditions.

5.1.2 Setup

We follow the experimental setup of (Saenko et al., 2010; Gong et al., 2012; Hoffman et al., 2013), which we summarize in the following. We have four domains (A, W, D, C) and the same 10 object categories per domain. For each experiment, one domain is selected as source domain and the other three as target domains. The number of examples per category varies from domain to domain and from category to category. When A is the source, 20 examples are randomly selected per category for training, while when the sources are either W, D or C, only 8 examples are selected per category. When a domain plays the role of target, only 3 examples are selected per category for performing the domain adaptation (training). All the examples of the target domains not used for training are used for testing. The accuracy of the

classification is measured as the number of correctly classified test examples divided by the total number of them (*i.e.*, without distinguishing object categories). In fact, since the splitting of the available examples into training (source and target) and testing (target) is based on random selection, each experiment is repeated 20 times. Therefore, the average of the 20 obtained accuracy values is actually used as final accuracy measure together with its associated standard deviation.

In order to make easier across-paper comparisons, we use the same 20 random train/test splits available from (Hoffman et al., 2013). Moreover, rather than using our own feature computation software, we use the pre-computed SURF-based bag of (visual) words (BoW) available for the images of *Office-Caltech*. Then, following (Gong et al., 2012), we apply PCA to such original SURF-BoW to obtain histograms of 20 visual words (bins).

5.1.3 Baselines

We compare our algorithm to the baselines summarized in Table 3. A-SVM, PMT-SVM, A-SSVM and A-SSVM-ALL are adapted with the target domain examples and the source classifiers, the rest of methods require the target domain examples and the original source domain ones for retraining. For the A-SVM and PMT-SVM methods we use the implementation provided by (Aytar and Zisserman, 2011), including MOSEK optimization (Mosek, 2013). We run GFK and MMDT using the code of (Hoffman et al., 2013). Note that in (Hoffman et al., 2013), GFK and MMDT are the best performing methods among others, including ARCT (Kulis et al., 2011) and HFA (Duan et al., 2012) methods. All these methods, except A-SSVM-ALL, follow the one-to-one domain adaptation style (Fig. 1(a)), *i.e.*, an independent domain adapation is performed for each target domain.

5.1.4 Results

We first evaluate the accuracy of HA-SSVM with a two-layer adaptation tree, *i.e.*, all the target domain datasets are at the same layer and connected to the source domain dataset by an intermediate node, similar to Fig. 2(a). The accuracy for each source/targets split is shown in Table 4. Table 5 shows the accuracy of each algorithm averaged over all domain splits. It is worth to note that our results for GFK and MMDT are totally in agreement with the ones presented in (Hoffman et al., 2013) for the same experiments and settings.

From Table 4 and Table 5, it is clear the importance of using all the available target-domain examples. Note that the best performing methods, A-SSVM-ALL (Fig. 1(b) style) and HA-SSVM (Fig. 1(c) style), do so in contrast to the rest of methods, which follow the one-to-one domain adaptation style (Fig. 1(a)). For instance, if we focus in the A→[W,D,C] case, both A-SSVM-ALL and HA-SSVM use 90 target domain examples simultaneously, *i.e.* 3 target domains × 10 object categories per target domain × 3 examples per category. 1-to-1 domain adaptation style methods use 30 W examples for performing the A→W domain adaptation, and analogously for A→D and A→C. Therefore, potential commonalities between W, D, and C domains are not used. Moreover, HA-SSVM outperforms A-SSVM-ALL, in agreement with our hypothesis that using the underlying hierarchical structure of the target domains is better than just mixing them blindly.

Focusing then on HA-SSVM, it is also interesting to see if other target domain structure (*e.g.*, a three-layer hierarchy) can improve the domain adaptation accuracy obtained so far. We test HA-SSVM with various three-layer adaptation trees. Table 6 shows the results. The three-layer adaptation tree achieves results as good as the ones of the two-layer tree

Adaptation Tree	A→W	A→D	A→C	Avg.
A→[W, D, C]	69.8 ± 0.7	59.7 ± 0.9	42.1 ± 0.4	48.4
A→[W, [D, C]]	69.8 ± 0.7	59.5 ± 1.0	40.1 ± 0.4	47.7
A→[D, [W, C]]	69.8 ± 0.6	59.1 ± 0.8	40.9 ± 0.4	47.8
A→[C, [D, W]]	72.7 ± 0.7	63.4 ± 1.2	42.1 ± 0.4	**49.5**
Adaptation Tree	W→A	W→D	W→C	Avg.
W→[A, D, C]	54.4 ± 0.6	66.1 ± 1.1	39.4 ± 0.3	47.3
W→[A, [D, C]]	54.3 ± 0.5	63.3 ± 1.3	38.7 ± 0.5	46.9
W→[D, [A, C]]	55.7 ± 0.6	65.8 ± 1.0	39.5 ± 0.4	**48.1**
W→[C, [A, D]]	54.2 ± 0.6	63.5 ± 1.0	39.5 ± 0.4	47.3
Adaptation Tree	D→A	D→W	D→C	Avg.
D→[A, W, C]	52.6 ± 0.5	73.0 ± 0.5	39.2 ± 0.6	48.7
D→[A, [W, C]]	52.6 ± 0.6	71.8 ± 0.7	39.4 ± 0.6	48.5
D→[W, [A, C]]	54.0 ± 0.6	73.0 ± 0.8	39.9 ± 0.6	**49.6**
D→[C, [A, W]]	53.0 ± 0.6	71.0 ± 0.7	38.9 ± 0.6	48.3
Adaptation Tree	C→A	C→W	C→D	Avg.
C→[A, W, D]	53.4 ± 0.8	69.6 ± 0.7	61.2 ± 0.9	57.3
C→[A, [W, D]]	53.2 ± 0.7	71.2 ± 0.7	63.0 ± 1.1	**57.8**
C→[W, [A, D]]	53.2 ± 0.6	69.5 ± 0.6	59.7 ± 1.3	57.1
C→[D, [A, W]]	52.4 ± 0.6	68.9 ± 1.0	61.0 ± 1.1	56.5

Table 6 HA-SSVM trained with various adaptation trees. The first column illustrates the tree structure. A two-layer adaptation tree is represented by X→[Y,Z,T], where X is the source domain and Y, Z and T are sibling target domains. These results are just a copy of the HA-SSVM ones shown in Table 4. A three-layer adaptation tree is represented by X→[Y,[Z,T]], where Z and T are siblings on the third layer and Y is located on the second layer.

	Amazon	DSLR	Webcam	Caltech256
Amazon	—	8.13	9.03	**9.78**
DSLR	8.13	—	**9.60**	8.25
WebCam	9.03	**9.60**	—	8.96
Caltech256	**9.78**	8.25	8.96	—

Table 7 Domain similarities in terms of QDS values (Ni et al., 2013). Lager values indicate higher similarity.

and some of them are even better. By further analyzing the domain relationships of the *Office* and *Caltech256* datasets, we found that there are strong connections to previous studies on domain similarity. In particular, to the rank of domain (ROD) (Gong et al., 2012) and the quantification of domain shift (QDS) (Ni et al., 2013). We show the domain similarities in Table 7 using QDS measurement. We note that the three-layer hierarchies which yield to best accuracies are those that best capture the underlying domain relationship. For instance, in the first group of Table 6, A→[C, [D, W]] achieves better accuracy than other adaptation trees, which is in agreement with the fact that [D, W] show higher similarity than [D, C] and [C, W] (see Table 7).

5.2 Latent Target Sub-Domains

Now we consider the scenario where the domain labels are not given a priori for the target data. In particular, we use again the *Office-Caltech* dataset with the same settings than in Sect. 5.1.2. However, we mix the target datasets by removing the domain labels. In these experiments, we first compare two recent domain discovery algorithms, in particular, latent domain discovery (Hoffman et al., 2012) (we call it *LatDD*), and domain reshaping (Gong et al., 2013b) (we call it *Reshape*). Finally, we evaluate the adaptation accuracies with the discovered domains, using HA-SSVM.

LatDD and *Reshape* require category labels to operate. However, in our domain adaptation setting we assume that only a few target domain examples have category label, which may be a handicap for such domain discovery methods. In this point, as proof-of-concept, we assumed that the target domain data does not have category labels. Therefore, we first applied the source domain model to predict the category labels in the unlabeled target domain (*i.e.*, the domain obtained by mixing the three domains not used as source). We denote by *LatDD-Pr* and *Reshape-Pr* the cases where we use predicted category labels instead of the groundtruth category labels.

LatDD requires as input the number of sub-domains to be discovered (originally this method has been developed to discover source domains), while *Reshape* involves an iterative process to search for the optimum number of sub-domains. We want to compare the HA-SSVM results in terms of discovered sub-domains *vs* a priori given ones, but only from the point of view of how the target data is distributed among a predefined number of target sub-domains. In other words, in these experiments we do not want the number of domains to be discovered. Therefore, we set this value to 3 for fair comparison with the experiments in Sect. 5.1. It is worth to note that for *Reshape/Reshape-Pr* we only use the so-called *distinctiveness* maximization step (Gong et al., 2013b).

Fig. 9 depicts the domain discovery results. It can be seen that *Reshape* and *Reshape-Pr* are clearly more accurate than *LatDD* and *LatDD-Pr* predicting the domains. Comparing *Reshape* and *Reshape-Pr*, we see that the former is more accurate as should be expected since it relies on groundtruth data. Comparing *LatDD* with *LatDD-Pr*, the accuracy differences are smaller than for *Reshape* and *Reshape-Pr*.

Now, for applying HA-SSVM, *LatDD-Pr* and *Reshape-Pr* are treated equally and as follows. For each discovered sub-domain, we assume that 3 examples are category-labeled for each category. Since our experiments are with 10 categories, as in Sect. 5.1, 90 target-domain examples must be available for performing domain adaptation (training) and the rest are used for testing. Since this train/test split is based on random selection, we repeat each experiment 20 times in order to emulate the setting of Sect. 5.1. Note that for *LatDD-Pr* and *Reshape-Pr* this means that we discard the predicted category labels, but we require only 90 examples to be labeled. In fact, *LatDD* and *Reshape* are not considered for HA-SSVM since these methods would require the category labels of all the target data (we included them in Fig. 9 just as reference to compare with their *predicted* counterparts).

Table 8 shows the final domain adaptation accuracies. As in Sect. 5.1 we evaluate two- and three-layer hierarchies, for the latter we only show the best obtained result among all possible configurations. We see that these results are comparable to the best obtained in Sect. 5.1 (also included in Table 8 as 'Given' for the reader convenience). Although we work with discovered sub-domains, HA-SSVM still outperforms the single-layer adaptation pooling-all strategy. *Reshape-Pr* outperforms *LatDD-Pr* as expected given the domain discovery accuracies seen in Fig. 9. However, the differences in accuracy are much larger for domain discovery than for the final object category classification, which may be due to the fact that HA-SSVM trains all the object category classifiers simultaneously for all domains in the hierarchy; thus, partially compensating domain assignment errors. Finally, for illustration purposes, Fig. 10 shows object examples within the domains discovered by *Reshape-Pr* and some of the three-layer adaptation trees used by HA-SSVM.

6 Domain Adaptation for Face Recognition

In this section, we apply HA-SSVM to the task of face recognition, where we aim to identify the person shown in a face-style image; thus it is similar to the task of multiple object recognition. We use Eq. (12) as the classifier. We evaluate domain adaptation algorithms on two different scenarios: (1) domain adaptation across pose variations, where the source and target domains correspond to different head poses; (2) domain adaptation across illumination variations, where the source and target domains contain face images with different

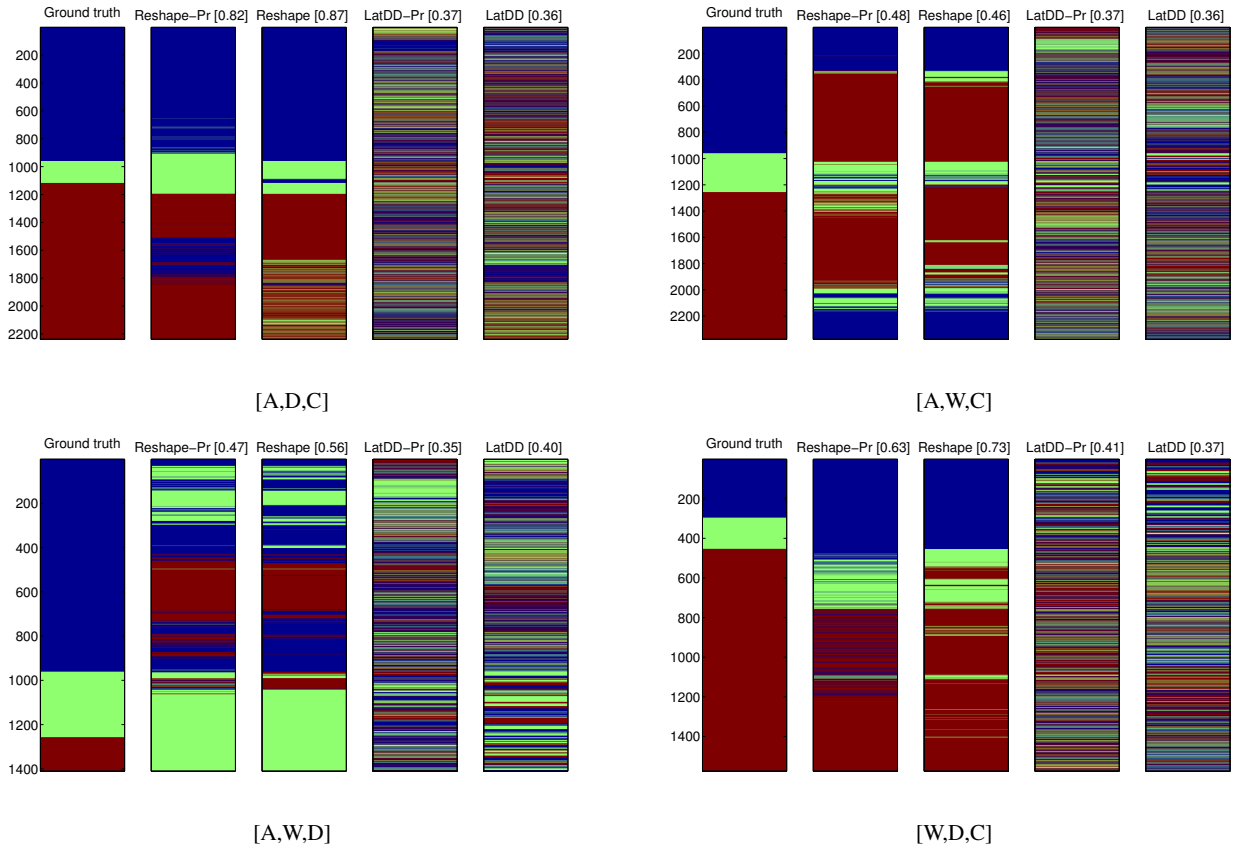

Fig. 9 Visualization of domain discovery results. The vertical axis indicates the indexes of the examples. Each color represents a different domain. For each sub-figure, the first column shows the original domain (groundtruth). The following columns are domain reshaping with predicted category labels ('Reshape-Pr'), domain reshaping with groundtruth category labels ('Reshape'), latent domain discovery with predicted category labels ('LatDD-Pr') and latent domain discovery with groundtruth category labels ('LatDD'). Within the brackets we show the estimated domain discovery accuracy running in [0,1].

Source Domain			A	W	D	C
Original Target Domains			W, D, C	A, D, C	A, W, C	A, W, D
Method	Hierarchy	Domain Discovery				
A-SSVM-ALL	—	—	47.6 ± 0.4	45.4 ± 0.4	46.5 ± 0.5	54.4 ± 0.6
HA-SSVM	2 Layers	Given	48.4 ± 0.5	47.3 ± 0.5	48.7 ± 0.6	57.3 ± 0.8
HA-SSVM	3 Layers	Given	**49.5** ± 0.4	**48.1** ± 0.6	**49.6** ± 0.5	57.8 ± 0.7
HA-SSVM	2 Layers	LatDD-Pr	46.2 ± 0.3	45.2 ± 0.4	45.9 ± 0.4	53.1 ± 0.6
HA-SSVM	3 Layers	LatDD-Pr	46.3 ± 0.4	45.1 ± 1.4	45.8 ± 0.4	53.0 ± 0.7
HA-SSVM	2 Layers	Reshape-Pr	49.0 ± 0.6	47.0 ± 0.5	48.0 ± 0.5	**59.4** ± 0.5
HA-SSVM	3 Layers	Reshape-Pr	49.1 ± 0.6	47.9 ± 0.5	48.2 ± 0.5	**59.1** ± 0.5

Table 8 The average multi-category recognition accuracy for a single target domain, for a priori 'Given' domains (Sect. 5.1), and for discovered latent domains (*LatDD-Pr* and *Reshape-Pr*). For the three-layer hierarchies we only show the best result among all possible three-layer adaptation trees.

illumination conditions. For each scenario, we investigate two types of features: (1) Similarly to the *Office-Caltech* dataset, we use SURF-BoW features (800 visual words of 128 dimensions); (2) As features learned by deep convolutional neural networks (CNN) has shown superior accuracy to the hand-crafted ones (*e.g.*, HOG, SURF), we also compute deep-learned features. Caffe deep learning framework (Jia et al., 2014) is used to extract 4096 dimensional features in our experiments.

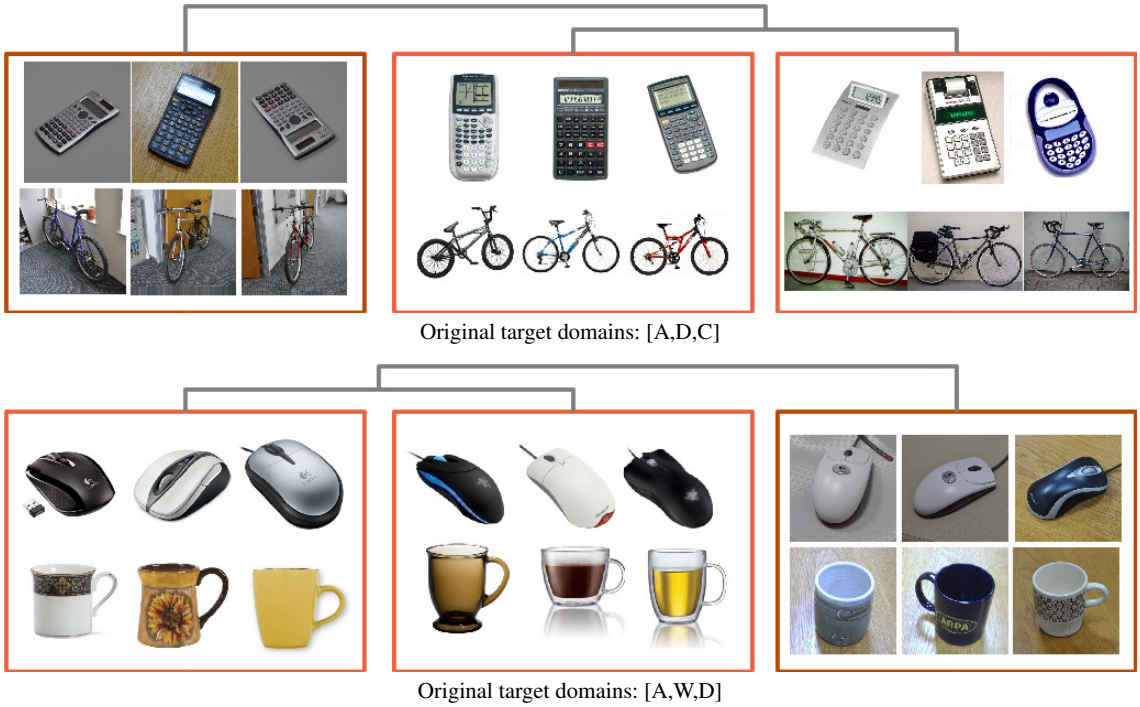

Fig. 10 *Reshape-Pr* + HA-SSVM qualitative results. Three exemplars for two categories are shown for each domain discovered by *Reshape-Pr*. The three-layer hierarchy used by HA-SSVM is also indicated for the underlying domains [A,D,C] (top) and [A,W,D] (bottom). In both cases, it correspond to the most accurate HA-SSVM-based multi-category classifier among the different ones that can be obtained for different three-layer hierarchy configurations.

6.1 Domain Adaptation Across Pose Variations

In this case, we use the head pose data in (Gourier et al., 2004)[5]. It consists of 15 different sets of images, each set corresponds to a different person. Globally there are persons with different skin color, and the same person appears with and without glasses. Each set contains images of the same person captured at various poses. The pose, or head orientation is determined by two angles (h, v), which varies from -90 degrees to $+90$ degrees. We split the dataset into different domains according to the horizontal angles h. To challenge domain adaptation, we select 3 domains which have large gap in horizontal angles. The selected angles and domains are illustrated in Table 9. For each target domain, we randomly select 3 examples per subject for DA training and the rest are used for testing. Since we repeat each type of experiment 5 times for obtaining a mean and standard derivation of the adaptation accuracy, such a random selection brings different adaptation-testing partitions per experiment. We build the two-layer hierarchical adaptation tree for HA-SSVM that is shown in Fig. 11. The results of the source and adapted classifiers are shown in Table 11 and Table 12, where HA-SSVM-N0 is the intermediate adaptation node.

Table 11 shows that using SURF features HA-SSVM outperforms HA-SSVM-N0 and A-SSVM-ALL, these two performing very similarly. Note that depending on the application we can either apply HA-SSVM-N0 (the root target-domain model) or HA-SSVM (the corresponding leaf target-domain model). For instance, if we are in an application where we cannot know a priori the horizontal angle range of the face, then we apply HA-SSVM-N0, while if we know it (*e.g.* the image is taken in a control point by a human and send it to a face recognition system, or different cameras capture different angles) then we can apply HA-SSVM.

[5] `www-prima.inrialpes.fr/Pointing04/data-face.html`

Domain	Source	T1	T2
Horizontal angle	0	-75,-90	75,90
# training examples per subject	20	3	3
# testing examples per subject	—	25	25

Table 9 Head pose data: horizontal angles and number of training and testing examples per subject in each domain. We have selected three very different pose ranges with respect to the horizontal angle (0°, $[-75°, -90°]$, $[75°, 90°]$), just for challenging more the domain adaptation methods.

Domain	Source	T1	T2
Illumination direction	Frontal	Left	Right
# training examples per subject	32	3	3
# testing examples per subject	—	13	13

Table 10 YaleB dataset: the illumination direction and the number of training and testing examples per subject in each domain.

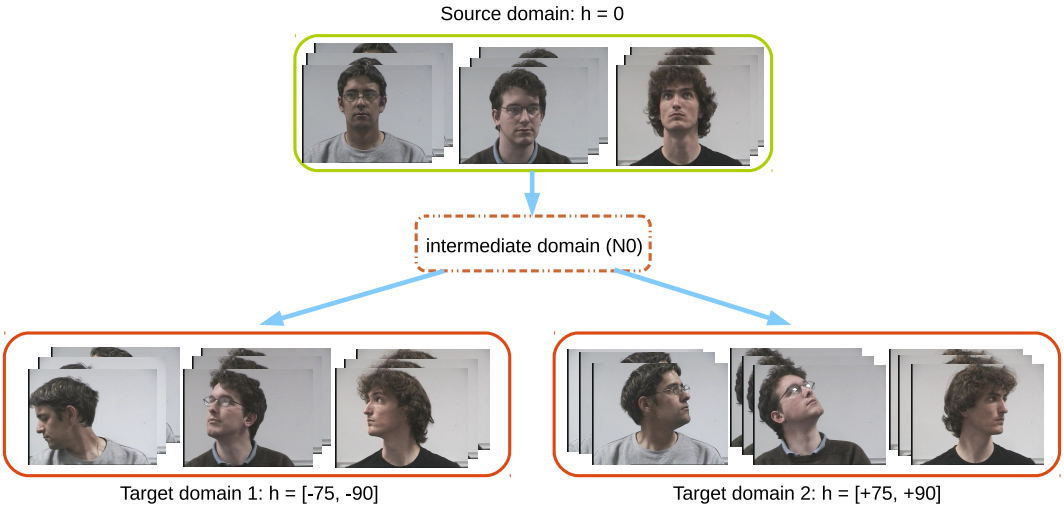

Fig. 11 Head pose dataset and the hierarchical adaptation tree. We split the dataset into 3 domains according to the horizontal angle of the head. The front view poses are used as source domain and the remaining two domains are used as target domains.

Test domain	SRC	A-SSVM-ALL	HA-SSVM-N0	HA-SSVM
Pose-T1	72.1 ± 0.2	88.9 ± 1.9	88.9 ± 1.6	**91.2** ± 1.1
Pose-T2	72.6 ± 0.9	88.4 ± 1.9	88.7 ± 1.4	**89.9** ± 1.6

Table 11 Head pose variation domain adaptation using SURF features, the average precision and standard deviations of the adapted classifiers. HA-SSVM-N0 is the intermediate adaptation node.

Test domain	SRC	A-SSVM-ALL	HA-SSVM-N0	HA-SSVM
Pose-T1	86.5 ± 0.3	95.9 ± 0.7	**96.2** ± 0.3	95.7 ± 1.1
Pose-T2	82.4 ± 0.4	97.8 ± 1.1	97.8 ± 1.2	**98.5** ± 0.4

Table 12 Head pose variation domain adaptation using Caffe features, the average precision and standard deviations of the adapted classifiers. HA-SSVM-N0 is the intermediate adaptation node.

Thus, in the worst case our method would perform as A-SSVM-ALL, while it is able to improve if target-domain information is available.

When using Caffe features (Table 12), the accuracy of all these classifiers are greatly improved and the differences of these DA algorithms become unclear. This is because, in this particular example, the domain shift is already largely reduced by the deep learned features and there is little space to improve.

6.2 Domain Adaptation Across Illumination Variations

In this case, we use the extended Yale Face dataset B (YaleB) (Georghiades et al., 2001; Lee et al., 2005). The YaleB dataset contains human subjects under 9 poses and 64 illumination conditions. We use the 10 subjects and split the dataset into three domains according to the illumination conditions, *i.e.* frontal, left side and right side lightening. The domain splits and the number of training and testing images are described in Table 10. Sample images are depicted in Fig. 12. Similar to the pose variation experiments, we build two layers hierarchical adaptation tree for HA-SSVM (see Fig. 12) and assess the accuracy of the intermediate adaptation node HA-SSVM-N0. Again, 5 rounds are executed per type of experiment, randomly selecting 3 target-domain images for the adaptation; then, the mean and standard derivation of the accuracy are computed as shown in Table 13 and Table 14.

Table 13 shows that using SURF features HA-SSVM-N0 outperforms A-SSVM-ALL, and HA-SSVM clearly outperforms them. The standard deviation of HA-SSVM and HA-SSVM-N0 accuracy is relatively high in this case. This indicates that in this experiments not all target-domain examples were equally useful (something that could be taking into account by using active learning as in (Vázquez et al., 2014)); however, in the individual experiments A-SSVM-ALL provided a worse accuracy than HA-SSVM and HA-SSVM-N0. Again, when using Caffe features (Table 14), the accuracy of all these classifiers is boosted so that the overall model is saturated and there is no room for domain adaptation in general.

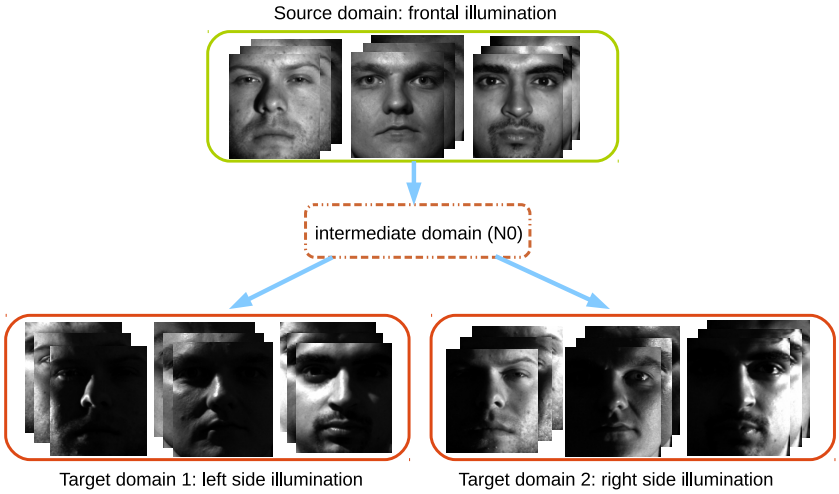

Fig. 12 YaleB dataset and the hierarchical adaptation tree. We split the dataset into 3 domains according to the illumination directions. The frontal illumination corresponds to the source domain and left and right illumination correspond to two target domains.

Test domain	SRC	A-SSVM-ALL	HA-SSVM-N0	HA-SSVM
YaleB-T1	20.3 ± 1.7	26.8 ± 2.0	38.2 ± 8.6	**40.0** ± 8.1
YaleB-T2	20.6 ± 1.3	23.5 ± 3.6	35.6 ± 8.5	**37.6** ± 9.6

Table 13 Illumination variation domain adaptation using SURF features, the average precision and standard deviations of the adapted classifiers. HA-SSVM-N0 is the intermediate adaptation node.

Test domain	SRC	A-SSVM-ALL	HA-SSVM-N0	HA-SSVM
YaleB-T1	52.3 ± 2.3	69.7 ± 3.0	**70.2** ± 3.8	**70.2** ± 5.3
YaleB-T2	53.6 ± 2.9	64.4 ± 3.4	**65.0** ± 3.7	**65.0** ± 5.9

Table 14 Illumination variation domain adaptation using Caffe features, the average precision and standard deviations of the adapted classifiers. HA-SSVM-N0 is the intermediate adaptation node.

7 Conclusions

In this paper, we present a novel domain adaptation method which leverages multiple target domains (or sub-domains) in a hierarchical adaptation tree. The key idea of the method is to exploit the commonalities and differences of the jointly considered target domains. Given the increasing interest on structural SVM (SSVM) classifiers, we have applied this idea to the domain adaptation method known as adaptive SSVM (A-SSVM), which only requires the target domain samples together with the existing source-domain classifier for performing the desired adaptation. Thus, in contrast with many other methods, the source domain samples are not required. Altogether, we term the presented domain adaptation technique as hierarchical A-SSVM (HA-SSVM).

As proof of concept we have applied HA-SSVM to pedestrian detection, object category recognition, and face recognition. The former involved to apply HA-SSVM to the deformable part-based model (DPM) while the others implied their application to multi-category classifiers. We showed how HA-SSVM is effective in improving the classification accuracy with respect to state-of-the-art strategies that ignore the structure of the target data. Moreover, focusing on the object category recognition application, we have evaluated HA-SSVM assuming that the target domains are discovered, obtaining comparable results to the case in which such domains are known a priori. In fact, we have seen that, thanks to such hierarchy of models, we can apply one sub-domain model or another (from the root to the leafs) depending on the target-domain a priori information that we have for a sample that must be classified.

As future work we would like to incorporate some recent advances in domain adaptation within the HA-SSVM framework. In particular, our structure-aware A-SSVM (SA-SSVM) approach (Xu et al., 2014a), as well as the cross-domain attribute codes (Mirrashed and Rastegar, 2013). Additionally, we want to explore how to incorporate this hierarchical adaptation withing deep architectures.

Acknowledgements This work is supported by the Spanish MEC project TRA2014-57088-C2-1-R, the Spanish DGT project SPIP2014-01352, the Generalitat de Catalunya project 2014-SGR-1506, Jiaolong Xu's Chinese Scholarship Council (CSC) grant No.2011611023, and Sebastian Ramos' FPI Grant BES-2012-058280. Finally, we also want to

thank the NVIDIA Corporation for the generous support in the form of different GPU hardware units.

References

A.Geiger, C.Wojek, and R.Urtasun (2011). Joint 3D estimation of objects and scene layout. In *Advances in Neural Information Processing Systems*, Granada, Spain.

Aytar, Y. and Zisserman, A. (2011). Tabula rasa: Model transfer for object category detection. In *Int. Conf. on Computer Vision*.

Behley, J., Steinhage, V., and Cremers, A. B. (2013). Laser-based segment classification using a mixture of bag-of-words. In *IEEE Int. Conf. on Intelligent Robots and Systems*.

Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., and Vaughan, J. (2009). A theory of learning from different domains. *Machine Learning*, 79(1):151–175.

Bergamo, A. and Torresani, L. (2010). Exploring weakly-labeled web images to improve object classification: a domain adaptation approach. In *Advances in Neural Information Processing Systems*, Vancouver, BC, Canada.

Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *IEEE Conf. on Computer Vision and Pattern Recognition*, San Diego, CA, USA.

Daumé III, H. (2007). Frustratingly easy domain adaptation. In *Meeting of the Association for Computational Linguistics*, Prague, Czech Republic.

Daumé III, H. (2009). Bayesian multitask learning with latent hierarchies. In *UAI*, Montreal, QC, Canada.

Dollár, P., Wojek, C., Schiele, B., and Perona, P. (2012). Pedestrian detection: an evaluation of the state of the art. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 34(4):743–761.

Duan, L., Tsang, I. W., Xu, D., and Chua, T.-S. (2009). Domain adaptation from multiple sources via auxiliary classifiers. In *Int. Conf. on Machine Learning*, Montreal, Quebec, Canada.

Duan, L., Xu, D., and Tsang, I. W. (2012). Learning with augmented features for heterogeneous domain adaptation. In *Int. Conf. on Machine Learning*, Edinburgh, Scotland.

Ess, A., Leibe, B., and Gool, L. V. (2007). Depth and appearance for mobile scene analysis. In *Int. Conf. on Computer Vision*, Rio de Janeiro, Brazil.

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

Finkel, J. and Christopher, D. (2009). Hierarchical bayesian domain adaptation. In *NAACL*, Colorado, USA.

Geiger, A., Lenz, P., and Urtasun, R. (2012). Are we ready for autonomous driving? the kitti vision benchmark suite. In *IEEE Conf. on Computer Vision and Pattern Recognition*, Washington, DC, USA.

Georghiades, A., Belhumeur, P., and Kriegman, D. (2001). From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23(6):643–660.

Girshick, R. (2012). *From Rigid Templates to Grammars: Object Detection with Structured Models*. PhD thesis, The University of Chicago, Chicago, IL, USA.

Girshick, R., Felzenszwalb, P., and McAllester, D. (2012). Discriminatively trained deformable part models, release 5. http://people.cs.uchicago.edu/ rbg/latent-release5/.

Gong, B., Grauman, K., and Sha, F. (2013a). Connecting the dots with landmarks: Discriminatively learning domain-invariant features for unsupervised domain adaptation. In *Int. Conf. on Machine Learning*, Atlanta, USA.

Gong, B., Grauman, K., and Sha, F. (2013b). Reshaping visual datasets for domain adaptation. In *Advances in Neural Information Processing Systems*, Lake Tahoe, NV, USA.

Gong, B., Grauman, K., and Sha, F. (2014). Learning kernels for unsupervised domain adaptation with applications to visual object recognition. *Int. Journal on Computer Vision*, 109(1-2):3–27.

Gong, B., Shi, Y., Sha, F., and Grauman, K. (2012). Geodesic flow kernel for unsupervised domain adaptation. In *IEEE Conf. on Computer Vision and Pattern Recognition*, Providence, RI, USA.

Gopalan, R., Li, R., and Chellappa, R. (2011). Domain adaptation for object recognition: An unsupervised approach. In *Int. Conf. on Computer Vision*, Barcelona, Spain.

Gourier, N., Hall, D., and Crowley, J. L. (2004). Estimating face orientation from robust detection of salient facial features. In *Int. Conf. in Pattern Recognition*.

Griffin, G., Holub, A., and Perona, P. (2007). Caltech-256 object category dataset. Technical report, California Institute of Technology.

Hoffman, J., Kulis, B., Darrell, T., and Saenko, K. (2012). Discovering latent domains for multisource domain adaptation. In *European Conf. on Computer Vision*, Florence, Italy.

Hoffman, J., Rodner, E., Donahue, J., Kulis, B., and Saenko, K. (2014). Asymmetric and category invariant feature transformations for domain adaptation. *Int. Journal on Computer Vision*, 109(1-2):28–41.

Hoffman, J., Rodner, E., Donahue, J., Saenko, K., and Darrell, T. (2013). Efficient learning of domain invariant image representations. In *Int. Conf. on Learning Representations*, Arizona, USA.

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., and Darrell, T. (2014). Caffe: Convolutional architecture for fast feature embed-

ding. *arXiv preprint arXiv:1408.5093*.

Jiang, J. (2008). A literature survey on domain adaptation of statistical classifiers. Technical report, School of Information Systems, Singapore Management University.

Kan, M., Wu, J., Shan, S., and Chen, X. (2014). Domain adaptation for face recognition: Targetize source domain bridged by common subspace. *Int. Journal on Computer Vision*, 109(1-2):94–109.

Kulis, B., Saenko, K., and Darrell, T. (2011). What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. In *IEEE Conf. on Computer Vision and Pattern Recognition*, Washington, DC, USA.

Lee, K., Ho, J., and Kriegman, D. (2005). Acquiring linear subspaces for face recognition under variable lighting. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 27(5):684–698.

Lu, B., Chellappa, R., and Nasrabadi, N. M. (2015). Incremental dictionary learning for unsupervised domain adaptation. In *British Machine Vision Conference*, Swansea, UK.

Mansour, Y., Mohri, M., and Rostamizadeh, A. (2008). Domain adaptation with multiple sources. In *Advances in Neural Information Processing Systems*, Vancouver, Canada.

Mirrashed, F. and Rastegar, M. (2013). Domain adaptive classification. In *Int. Conf. on Computer Vision*, Sydney Australia.

Mosek (2013). Optimization toolkit. http://www.mosek.com.

Nguyen, H., Ho, H. T., Patel, V., and Chellappa, R. (2015). Dash-n: Joint hierarchical domain adaptation and feature learning. *IEEE Trans. on Image Processing*, 24(12):5479–5491.

Ni, J., Qiu, Q., and Chellappa, R. (2013). Subspace interpolation via dictionary learning for unsupervised domain adaptation. In *IEEE Conf. on Computer Vision and Pattern Recognition*, Oregon, USA.

Pan, S. and Yang, Q. (2009). A survey on transfer learning. *IEEE Trans. on Knowledge and Data Engineering*, 22(10):1345–1359.

Park, D., Ramanan, D., and Fowlkes, C. (2010). Multiresolution models for object detection. In *European Conf. on Computer Vision*, Crete, Greece.

Pepikj, B., Stark, M., Gehler, P., and Schiele, B. (2015). Multi-view and 3d deformable part models. *IEEE Trans. on Pattern Analysis and Machine Intelligence*.

Premebida, C., Carreira, J., Batista, J., and Nunes, U. (2014). Pedestrian detection combining rgb and dense lidar data. In *IEEE Int. Conf. on Intelligent Robots and Systems*, Chicago, IL, USA.

Saenko, K., Hulis, B., Fritz, M., and Darrel, T. (2010). Adapting visual category models to new domains. In *European Conf. on Computer Vision*, Hersonissos, Heraklion, Crete, Greece.

Tang, K., Ramanathan, V., Fei-fei, L., and Koller, D. (2012). Shifting weights: Adapting object detectors from image to video. In *Advances in Neural Information Processing Systems*, Lake Tahoe, Nevada, USA.

Vázquez, D., López, A., Marín, J., Ponsa, D., and Gerónimo, D. (2014). Virtual and real world adaptation for pedestrian detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 36(4):797–809.

Vázquez, D., López, A., and Ponsa, D. (2012). Unsupervised domain adaptation of virtual and real worlds for pedestrian detection. In *Int. Conf. in Pattern Recognition*, Tsukuba, Japan.

Xu, H., Zheng, J., and Chellappa, R. (2015). Bridging the domain shift by domain adaptive dictionary learning. In *British Machine Vision Conference*, Swansea, UK.

Xu, J., Ramos, S., Vázquez, D., and López, A. (2014a). Domain adaptation of deformable part-based models. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 36(12):2367–2380.

Xu, J., Vázquez, D., López, A., Marín, J., and Ponsa, D. (2014b). Learning a part-based pedestrian detector in a virtual world. *IEEE Trans. on Intelligent Transportation Systems*, 15(5):2121–2131.

Yang, J., Yan, R., and Hauptmann, A. (2007). Cross-domain video concept detection using adaptive SVMs. In *ACM Multimedia*, Augsburg, Germany.

Yebes, J., Bergasa, L., and García, M. (2015). Visual object recognition with 3d-aware features in kitti urban scenes. *Sensors*, 15(4):9228–9250.

Zhu, L., Chen, Y., Yuille, A., and Freeman, W. (2010). Latent hierarchical structural learning for object detection. In *IEEE Conf. on Computer Vision and Pattern Recognition*, San Francisco, CA, USA.