Article:
Htike, KK and Hogg, DC orcid.org/0000-0002-6125-9564 (2016) Adapting pedestrian detectors to new domains: A comprehensive review. Engineering Applications of Artificial Intelligence, 50. pp. 142-158. ISSN 0952-1976

https://doi.org/10.1016/j.engappai.2016.01.029

Adapting Pedestrian Detectors to New Domains: A Comprehensive Review

Kyaw Kyaw Htike

UCSI University
Kuala Lumpur, Malaysia

David Hogg

University of Leeds
Leeds, UK

Abstract

Successful detection and localisation of pedestrians is an important goal in computer vision which is a core area in Artificial Intelligence. State-of-the-art pedestrian detectors proposed in literature have reached impressive performance on certain datasets. However, it has been pointed out that these detectors tend not to perform very well when applied to specific scenes that differ from the training datasets in some ways. Due to this, domain adaptation approaches have recently become popular in order to adapt existing detectors to new domains to improve the performance in those domains. There is a real need to review and analyse critically the state-of-the-art domain adaptation algorithms, especially in the area of object and pedestrian detection. In this paper, we survey the most relevant and important state-of-the-art results for domain adaptation for image and video data, with a particular focus on pedestrian detection. Related areas to domain adaptation are also included in our review and we make observations and draw conclusions from the representative papers and give practical recommendations on which methods should be preferred in different situations that practitioners may encounter in real-life.

Keywords: domain adaptation; pedestrian detection; feature learning; scene-specific detector; transfer learning

1. Introduction

Due to the fact that visual perception is vital to most intelligent life forms, *computer vision* has become one of the most important and active research areas in the field of Artificial Intelligence. Computer vision is about automatic analysis and understanding of visual data (such as images and videos) to extract useful information.

There are many sub-areas within the field of computer vision, one of which is object detection which forms the foundation of many intelligent scene understanding systems. Due to its significance, object detection has received a lot of attention in computer vision [1].

Pedestrian detection in particular plays an important role in real world outdoor scenes, especially in urban areas. Although many proposed domain adaptation algorithms in literature could potentially be used for learning detectors for a variety of different object categories (such as pedestrians, cars, buses and bicycles), we focus on the task of domain adaptation for pedestrian detection since pedestrians are of most interest in many applications of computer vision.

2. Motivation

Pedestrian detection in monocular images is a challenging task and a lot of progress has been made in this area [2, 3, 4]. Most state-of-the-art pedestrian detectors require a supervised training stage based on a labelled dataset that is obtained from manual annotation of pedestrians (*e.g.* delineation of pedestrians by bounding boxes) and a sufficient number of non-pedestrian images [5, 2, 6].

2.1. Training & Generalisation

The objective of the labelled dataset is to provide the classifier (being learnt) with large intra-class variations of pedestrians and non-pedestrians so that the resulting classifier is *generalisable* to never-before-seen test data. This *generalisation* property is a sought-after property for most machine learning classification and regression tasks.

When training a pedestrian detector, the goal is often: "Given *any* unseen test image, the detector should locate all the pedestrians in the image". In this paper, we term such a detector as a *generic (pedestrian) detector* and the training data from which the detector was trained as a *generic (pedestrian) dataset*.

2.2. Generic Datasets

For a generic dataset, collected positive and negative examples are not (deliberately) limited to a particular scene and viewpoint and the aim of such a dataset is to collect as many variations of pedestrians as possible to produce detectors which should ideally perform well for any unseen test data. Examples of generic pedestrian datasets are INRIA Person Dataset [5], Daimler Mono Pedestrian Detection Benchmark Dataset [3]

and Caltech Pedestrian Dataset [7]. The INRIA dataset consists of images of upright people taken from a variety of personal image collections. Pedestrian training data of the Daimler and the Caltech datasets are extracted from videos recorded with on-board cameras in vehicles being driven around various places in urban traffic. All these datasets consist of training data from a variety of scenes and places, and as a result, the intra-class variations of pedestrians in such datasets is large. Figure 1 illustrates this observation.

2.3. Problems with Generic Datasets

Despite the large intra-class variations present in such generic datasets, each of these datasets still has its own inherent bias. For example, since the INRIA dataset is taken from mostly personal digital image collections, many of the people in the training dataset are likely to be intentionally posing for cameras. This may be different from natural pedestrian poses and activities in real-life situations. For the Daimler and Caltech datasets, the pedestrians in the training set are biased to view-points and angles that cameras on-board vehicles could capture. Moreover, pedestrians from these datasets are taken from static images that have been captured using cameras fixed near the same ground plane as the captured pedestrians. This may be considerably different from situations where images of pedestrians are captured by video cameras looking down on a scene (e.g. surveillance videos).

2.4. Dataset Bias

This dataset bias has been recently studied by Torralba and Efros [8]. No dataset can possibly cover a *representative* set of all the possible variations of pedestrians and non-pedestrians the detector is likely to face at test time. As shown by [7, 9], detectors may fail to perform satisfactorily when applied to scenes that differ from the original training data in many aspects such as:

- Pedestrian pose
- Image or video resolution
- View-point
- Lighting condition
- Image or video compression effects
- Presence of motion blur

2.5. Non-trivial Nature of Classifier Training

Furthermore, apart from this dataset problem, even assuming that there is a perfect generic dataset, it is non-trivial to learn a classifier that is "good" enough to capture all these highly complex and multi-modal variations of the dataset whilst at the same time not over-fit on the training data. In addition, for most of the pedestrian detectors, the speed of the detector is an important criterion that has to be taken into consideration, particulary since a classifier must be applied on many (typically millions of) multi-scale sliding windows in each image. This rules out time-consuming feature extraction mechanisms and complex classifiers.

2.6. Scene-specific Detectors

It is, however, crucial to ask the question of whether, deployed pedestrian detectors in real-life actually need to work well across any test data. The short answer is that for most situations, they do not. Each deployed pedestrian detector needs to work well only for the specific scene and conditions that it is applied to.

Given a particular scene, the intra-class variation of the pedestrians being captured by a fixed camera is limited compared to general situations. For example, due to the fixed camera angle, the view-point is fixed and the space of possible poses that a pedestrian can exhibit is a small subset of all the possible pedestrian poses. Furthermore, the lighting variation is also smaller and the image compression effects are similar for all the pedestrians captured by the same camera. In addition, the environment, the background and the surroundings are fixed which translate to less variation in non-pedestrian classes of data. Moreover, there are geographical, cultural and social constraints under a fixed location which, for example, may make pedestrians more likely to conform to similar styles of clothing.

Further, most state-of-the-art detectors work by extracting features from a rectangular window and applying the learnt classifier. This means that pixels that do not correspond to pedestrians (also known as "scene context") are also inside the window. For a particular scene, this scene context can be captured effectively. Overall, the intra-class variation of pedestrians (or non-pedestrians) in a specific scene is smaller than the intra-class variation of pedestrians (or non-pedestrians) across all possible scenarios as shown in Figure 2.

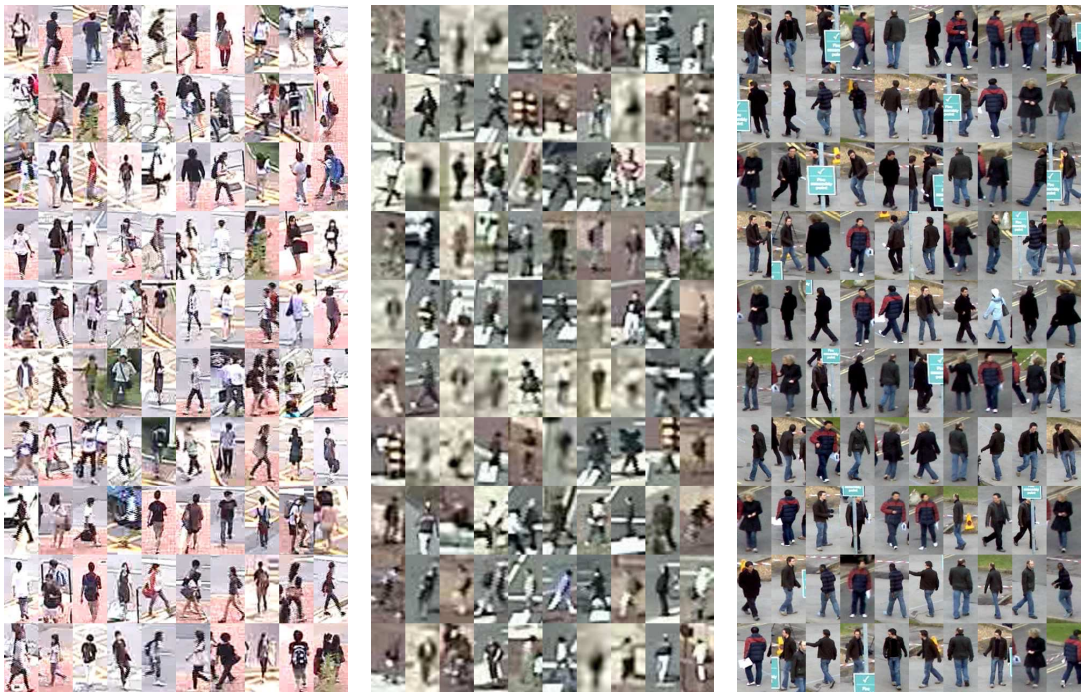
Therefore, it seems that the solution then is to collect labelled data for each new scene specifically tailored for that scene. The resulting detector can be termed as a *scene-specific detector* since the detector is tuned and specialised to work well in a particular type of scene. The task is now clearly simpler: given any unseen test image in *this* scene, the detector should locate all the pedestrians in the image. This is a simpler task than building a generic detector.

There are two observations that can be made. Firstly, with the feature extraction mechanism and the classifier type held fixed, a scene-specific detector can be more accurate than a generic detector. Secondly, with the detector accuracy held fixed, the feature extraction mechanism and the classifier of the scene-specific detector can be simpler and faster than a generic detector due to having to learn to perform classification for a simpler task. This is clearly critical in real-time or embedded-processor applications. In this paper, we survey methods that makes use of the first observation.

Although training a scene-specific detector that is specialised to each new scene seems like a good idea, in practice, it can be labour-intensive especially when considering the number of different scenes and applications for which we need pedestrian detectors. In this paper, we survey papers that address this problem by domain adaptation techniques that reduce the human supervision effort involved in learning scene-specific pedestrian detectors.

(a) INRIA dataset [5] (b) Daimler dataset [3] (c) Caltech dataset [7]

Figure 1: Random samples from some generic pedestrian datasets (only pedestrians, *i.e*. positive examples, are shown).



(a) CUHK Square dataset [10] (b) MIT Traffic dataset [11] (c) PETS 2009 dataset[12]

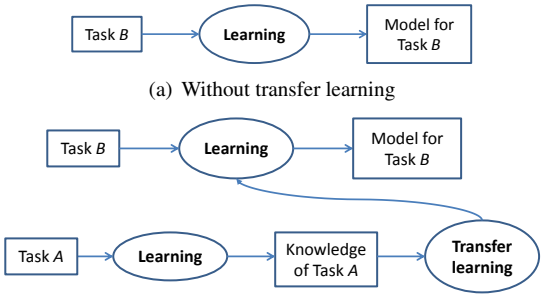
Figure 2: Random samples from scene-specific pedestrian datasets (only pedestrians, *i.e*. positive examples, are shown).

3. Background

In this section, we give an introduction to the basic concepts of transfer learning and domain adaptation that are needed to understand the terms used in this paper.

3.1. Transfer Learning

Stated informally, the concept of transfer learning, in the field of machine learning, is mainly relevant when we have *related* tasks, and knowledge about some of those tasks; having knowledge about some tasks can be used to learn about other related tasks in an easier, faster or improved manner. This is useful because, for many tasks in machine learning, we may have a large amount of labelled data for a task A but may not have sufficient labelled data (or even no labelled data) for a task B which is related to task A in some way. Using transfer learning, we can transfer the knowledge that we have about task A to task B using some commonality between task A and task B. This is illustrated in Figure 3.



(a) Without transfer learning

(b) With transfer learning

Figure 3: Transfer learning between two tasks. Given two related tasks A and B, exploiting and transferring the knowledge about task A to task B can help learn a better model for task B.

Another useful benefit of transfer learning is when deploying trained models (*e.g.* classifiers) for prediction at *test* time in real-life systems. There is one assumption common to most machine learning algorithms: the distribution and the feature space of the training data are the same as those of the test data. Generally, if the *feature space* of the test data is different from that of the training data, the current model cannot be applied to the test data and a new model would have to be trained in the feature space that is the same as the test data. If the feature spaces of the training and the test data are the same but the *distributions* of the training and the test data are different, the model that was trained on the training data may perform poorly on the test data depending on the extent of the difference between the training and test data distributions. If this difference is large enough, the model might not even give any meaningful predictions and a new model would then need to be trained.

Having to train new models in this way can be computationally expensive and with traditional machine learning methods, this is usually necessary because for many deployed machine learning systems, the test data distributions are different from those of the training data. Transfer learning can help here to a certain extent by considering the training and test data as data for two related tasks.

We now discuss two commonly used terms in the transfer learning literature [13]: a *domain* and a *task*. Furthermore, when discussing transfer learning below, we would do it in the context of *classification* although transfer learning can also be used for regression and density estimation.

A *domain* $\mathcal{D}$ is defined by a feature space $\mathcal{X}$ and a probability distribution $P(\mathbf{x})$ over the data associated with $\mathcal{D}$. A *task* $\mathcal{T}$ is specified by a label space $\mathcal{Y}$ and a distribution over the label space $P(y)$.

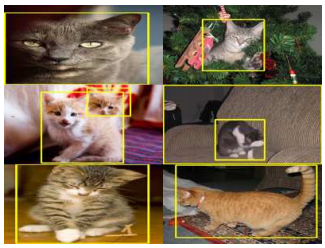
To give an example of a domain and a task, consider training a pedestrian classifier using the INRIA dataset. To simplify the explanation, assume that we are given cropped patches of pedestrians and non-pedestrians (also referred to as *positive patches* and *negative patches* respectively). For each patch, we extract features using any feature extraction algorithm to obtain a feature vector. The feature extraction mechanism defines the feature space $\mathcal{X}$. For instance, if we are using the HOG feature extraction algorithm which results in feature vectors of length 4608 and each dimension of a feature vector is a real number within the range of $[0, 0.2]$, then $\mathcal{X}$ is a 4608-dimensional space for which the values of each dimension has the range $[0, 0.2]$. After extracting features from each patch, we now have the training data $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ where the N number of training data are samples from the underlying distribution $P(\mathbf{x})$, *i.e.* $\mathbf{X} \sim P(\mathbf{x})$. For classification, each training datum $\mathbf{x}_i \in \mathbf{X}$ is also associated with a label y_i. There are N labels $\mathbf{Y} = \{y_1, \ldots, y_N\}$ for the training data. For pedestrian classification, the label is either *pedestrian* or *non-pedestrian*, *i.e.* $y_i \in \{\text{pedestrian}, \text{non-pedestrian}\}$. The training data $\mathbf{X}$ together with the labels $\mathbf{Y}$ is usually called a *labelled (training) dataset*. After obtaining the labelled dataset, a classifier can be trained using a supervised machine learning algorithm which produces a model that can be written as a function of $\mathbf{x}$: it takes in a feature vector $\mathbf{x}$ as input and produces a classification score as output. This can also be probabilistically interpreted as $P(y|\mathbf{x})$, *i.e.* the (posterior) probability of the class labels y given a feature vector $\mathbf{x}$. In summary, the domain and the task for this pedestrian classification setting is given by $\mathcal{D} = \{\mathcal{X}, P(\mathbf{x})\}$ and $\mathcal{T} = \{\mathcal{Y}, P(y)\}$ respectively.

	Source	Target
Domain	$\mathcal{X}_s, P(\mathbf{x}_s)$	$\mathcal{X}_t, P(\mathbf{x}_t)$
Task	$\mathcal{Y}_s, P(y_s)$	$\mathcal{Y}_t, P(y_t)$

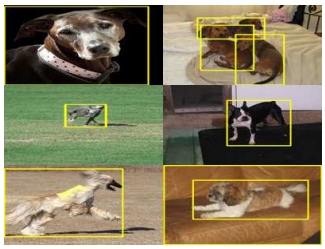
Table 1: Annotation summary for transfer learning

The annotation summary for transfer learning is given in Table 1. Given a *source domain* $\mathcal{D}_s = \{\mathcal{X}_s, P(\mathbf{x}_s)\}$ and a *source task* $\mathcal{T}_s = \{\mathcal{Y}_s, P(y_s)\}$, the aim of transfer learning is to transfer the "knowledge" in $\mathcal{D}_s$ and $\mathcal{T}_s$ to a *target domain* $\mathcal{D}_t = \{\mathcal{X}_t, P(\mathbf{x}_t)\}$ and a *target task* $\mathcal{T}_t = \{\mathcal{Y}_t, P(y_t)\}$ so that the learning of $P(y_t|\mathbf{x}_t)$ is improved. Although multiple sources and targets can be used for transfer learning, in this paper, we focus only on one source and one target since this situation is prevalent in real-life situations.

As an example application of transfer learning, consider a scenario in which datasets of cats and dogs are given (shown



(a) Samples from cat dataset



(b) Samples from dog dataset

Figure 4: Some samples of cats and dogs from the PASCAL VOC dataset [14]. Localisation of cats and dogs is shown with bounding boxes.

in Figure 4) and the source task is the detection of cats and the target task is the detection of dogs. We would like to exploit the knowledge that we have about cats (assuming the availability of a large amount of labelled data for cats) and *transfer* it to the process of learning a dog classifier (assuming insufficient labelled dog data) to help obtain a better dog classifier. This is possible because even though detection of cats and detection of dogs are different tasks, they are related in that cats and dogs share some similarities in appearance, shape and structure (as can be seen in Figure 4). The feature spaces of cats and dogs, $\mathcal{X}_s$ and $\mathcal{X}_t$ respectively, may or may not be the same, but $P(\mathbf{x}_s)$ and $P(\mathbf{x}_t)$ would be different. In addition, the label spaces of the source and target tasks, $\mathcal{Y}_s$ and $\mathcal{Y}_t$ respectively, are also different since $\mathcal{Y}_s = \{\text{cat}, \text{non-cat}\}$ and $\mathcal{Y}_t = \{\text{dog}, \text{non-dog}\}$ and $\mathcal{Y}_s \neq \mathcal{Y}_t$.

The example given above is a problem of *supervised transfer learning* because there is some labelled data available from the target dataset. An alternative setting is *unsupervised transfer learning* where there is no labelled data available from the target dataset.

3.2. Domain Adaptation

We now discuss a special case of transfer learning in which the source and target *tasks* are the *same* (*i.e.* $\mathcal{Y}_s = \mathcal{Y}_t$ and $P(y_s) = P(y_t)$) and the source and target *domains* are *different*. Moreover, even though the domains are different, the feature spaces of the source and target are the same (*i.e.* $\mathcal{X}_s = \mathcal{X}_s$ and $P(\mathbf{x}_s) \neq P(\mathbf{x}_t)$). This is known as *domain adaptation* which is actually a type of *transductive transfer learning* [13] and is simpler than the general transfer learning setting.

The reason for highlighting this particular form of transfer learning is that it can efficiently tackle the type of problem that we are interested in, which is adapting pedestrian detectors trained on a generic dataset (*e.g.* INRIA pedestrian dataset) to a specific scene (*e.g.* a surveillance video camera recording a traffic junction). This can be placed in a domain adaptation framework by assuming that the source domain is the data from the generic pedestrian dataset and the target domain is data that can be obtained from the specific scene. To show that this is a domain adaptation setting, the following observations can be made:

1. $\mathcal{X}_s = \mathcal{X}_t$: The source feature space is the same as the target feature space. This is because it is assumed that the same feature extraction mechanism is used for both the generic dataset and the specific scene.

2. $\mathcal{Y}_s = \mathcal{Y}_t$: The source label space is the same as the target label space. This is because for both the generic dataset and data from the specific scene, the label space is given by $\{\text{pedestrian}, \text{non-pedestrian}\}$.

3. $P(y_s) = P(y_t)$: The (prior) distributions on the labels for the generic dataset and specific scene are assumed equal.

4. $P(\mathbf{x}_s) \neq P(\mathbf{x}_t)$: The pedestrian distribution of the generic dataset is *not* the same as that of the specific scene. This is due to differences in image resolutions, illumination, pedestrian poses, camera angles, motion blur, *etc*. Even though $P(\mathbf{x}_s) \neq P(\mathbf{x}_t)$, there is still some relation between $P(\mathbf{x}_s)$ and $P(\mathbf{x}_t)$, and $P(\mathbf{x}_t)$ can be considered as an (unknown) transformation of $P(\mathbf{x}_s)$.

As can be seen, this is exactly a domain adaptation setting where the tasks for the source and target are the same and domains are different. Many computer vision problems can be placed in this domain adaptation framework.

Another example scenario where domain adaptation may be helpful is when having a face detector that is trained using a generic face dataset (such as the Faces in the Wild dataset [15] which contains over ten thousand images of faces collected from the Internet) and wanting to apply the detector to images taken in a more specific and controlled environment (such as the Yale Face Database [16]). Random samples from these datasets are shown in Figure 5. Although the face detector trained on the generic dataset may work reasonably well on the target dataset, it is expected that adapting the detector to specialise it to the target dataset (which may have much less intra-class variation of faces) might improve the detection performance in the target dataset. This is a domain adaptation problem because the feature spaces of the source and target datasets are the same since faces are represented by the same feature extraction mechanism (such as HOG or Haar features). Moreover, the tasks are the same since the label spaces and the prior label probabilisties are the same (both aims at face/non-face classification).

As with transfer learning, there are two main types of domain adaptation. In both types, we assume that we have a sufficiently large number of labelled data for the source dataset. The first type is *unsupervised domain adaptation*. In this type, we do not have any labelled data in the target dataset. In the second type, we assume that we have some labelled data in the target domain. This is known as *supervised domain adaptation*. In

(a) Face samples from Faces in the Wild dataset [15]

(b) Face samples from Yale Face Database [16]

Figure 5: Samples of faces from source and target domains. Note that for Yale dataset (on the right), only greyscale images could be obtained. Therefore, both of these datasets are shown in grayscale.

this paper, we are concerned with only unsupervised domain adaptation, which is a more difficult problem.

4. Survey

Domain adaptation is a relatively new research area and a fundamental topic in Artificial Intelligence. Early works on domain adaptation were published in the field of text and Natural Language Processing (NLP) [17, 18, 19, 20].

Hwa [20] proposes an adaptation approach for grammar structure induction using sparsely annotated training data (*i.e*. data with limited constituent information) to obtain results that are almost as good as using a fully annotated textual corpus.

Roark and Bacchian [19] make use of a maximum a posterior framework to adapt probabilistic context-free grammars to new domains. McClosky *et al*. [18] propose a parser adaptation system using self-training and re-ranking.

Blitzer *et al*. [17] propose an unsupervised domain adaptation for part-of-speech tagging by projecting the source dataset to a real-valued low-dimensional feature representation that is shared across the source and the target domains. This representation is learnt using structural correspondence learning which works by firstly defining a set of pivot features. Pivot features are frequently-occurring features that are invariant and discriminative across both domains. Secondly, correspondences among features of source and target domains are learnt with the help of these pivot features. Their proposed algorithm assumes that features from the domains are binary and also requires defining pivot features, which is not trivial especially in applications other than text.

In fact, most of the algorithms used for domain adaptation for NLP are not suitable for vision applications. Therefore, in this paper, we will focus on prior work about domain adaptation for computer vision rather than NLP.

Research for domain adaptation for vision is even more recent than NLP. There are mainly two areas of research in domain adaptation for vision: image classification and object detection. For image classification, the majority of the approaches for domain adaptation turn out to be metric learning or feature projection approaches.

Object detection is a harder and a more general task than image classification. Similarly, domain adaptation for object detection is generally a more challenging problem than domain adaptation for image classification. Some of these challenges are:

1. Extreme class imbalance: Object detection involves having to model the positive and negative class. The number of data in the negative class is much larger than that of the positive class and the positive class can easily get swamped with the negative class.

2. The adaptation algorithm not only has to deal with the intra-class variation of the positive class but also the much larger "sea" of intra-class variation of the negative class.

3. Due to object detection having different requirements as compared to image classification (such as needing to evaluate hundreds of thousands of candidate windows), object detection systems usually use a different set of features (such as Histogram of Oriented Gradients or Haar features) than image classification systems (which tend to use features such as "Bag of Visual Words"). Object detectors typically use very high dimensional and dense features whereas many image classification systems tend to use lower-dimensional and sparser features. This makes a difference in the required domain adaptation techniques. For example, generative models may be suitable for domain adaptation for image classification but ill-suited for domain adaptation for object detection.

It is therefore *not* straightforward or trivial to apply existing domain adaptation methods for image classification to the task of object detection.

Furthermore, domain adaptation for object detection in *videos* brings with it a unique set of challenges and opportunities which is different from that of image classification or even object detection in static images. Some of these opportunities include availability of spatio-temporal smoothness and other types of information that can be learnt and exploited from videos. This means that even if adopting existing domain adaptation techniques for image classification for object detection is easy, it may be more desirable to research and develop algorithms that exploit these cues in the videos for improved performance. Moreover, for far-field videos, pedestrians are of small resolution which further increases the challenge of domain adaptation. Therefore it is crucial to differentiate domain adaptation approaches for image classification from those for object detection (especially in videos).

Due to its unique challenges and opportunities, it turns out that different variations of *self-training* is the most popular approach for state-of-the-art domain adaptation for object detec-

Figure 6: The need for domain adaptation for image classification. Figure taken from [21].

tion in video. The popularity is due to the fact that the self-training framework is flexible, can work with a variety of discriminative classifiers and can incorporate different types of prior knowledge in a natural and easy way. For object detection, we will mainly focus on domain adaptation of object detectors trained on image datasets to videos.

4.1. Structure of Survey

We begin by discussing research related to domain adaptation for image classification (Section 4.2). Then we review domain adaptation for object detection for videos in Section 4.3. For the sake of completeness, in Section 4.4, this is followed by reviewing three areas that are not directly relevant but somewhat related to the topic of domain adaptation:

- Learning moving object detectors in videos (Section 4.4.1): In this section, we discuss approaches that learn class-agnostic moving object detectors in videos.
- Semi-supervised learning for object detection in videos (Section 4.4.2): Here, we review algorithms that learn object detectors in videos using semi-supervised learning (given a small amount of labelled data in the target domain with no notion of a source domain).
- Weakly-supervised learning of object detectors (Section 4.4.3): Methods based on weaker form of supervision than the standard (bounding box) supervision are discussed. Here, there is no true concept of source and target datasets. With weakly supervised learning, for a specific scene, only the target dataset can be considered to be present. The prior information comes from weak supervision on the target scene rather than in the form of a source dataset.

4.2. Domain Adaptation for Image Classification

The need for domain adaptation for image classification is illustrated in Figure 6. Most of the research in this area is based on learning a common feature representation across the source and target domains.

One of the earliest domain adaptation approaches for image classification is the work by Saenko *et al*. [21]. They provide a supervised domain adaptation algorithm that learns a regularised non-linear transformation that is invariant across the source and target domains. Learning such a transformation allows modelling of changes resulting from the difference in the

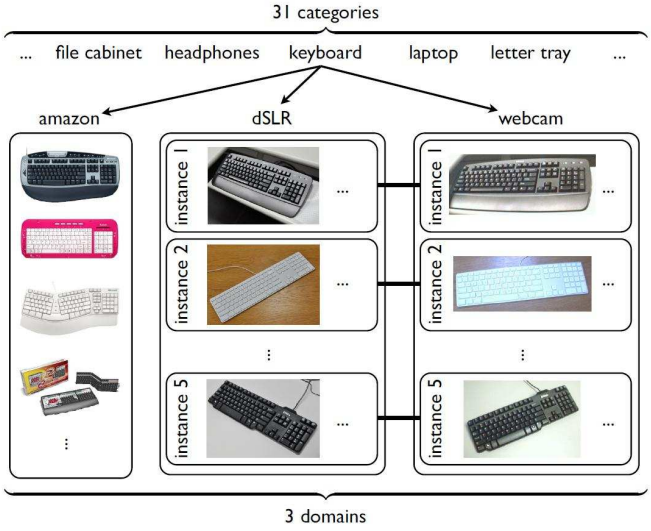


Figure 7: Multi-domain object database to study and evaluate domain adaptation algorithms for image classification proposed by [21]. The database contains 31 object categories and for each category, there are 3 domains: images taken from `Amazon.com`, a high resolution digital SLR camera and a simple low resolution webcam. Figure taken from [21].

source and target domains. They also introduce a multi-domain object database (shown in Figure 7) to evaluate domain adaptation algorithms for image classification. Their method requires exact manual mapping of samples from the source and target domains which can be very time-consuming.

An extension of [21] is proposed by Kulis *et al*. [22]. This time, instead of learning a single transformation as in [21], the authors propose learning asymmetric linear transforms, *i.e*. two linear transformations: one for the source domain and the other for the target domain, to respectively project the source and target data to a common subspace. In order to deal with non-linear asymmetric transformations, they kernalize the algorithm (by running the algorithm in the kernel space instead of the original feature space). Their approach however shares the same limitation as [21]: they require manual specification of pairs of source and target data examples that are similar semantically (*e.g*. two very similar cups (or even the same cup) taken from the source and target domains may form such a pair).

Before going further, we digress to provide a brief explanation on the concept of geodesic subspaces. Euclidean geometry, which many people are familiar with, is about flat spaces which can be demonstrated by drawing on a whiteboard. In the Euclidean space, a number of properties are preserved, such as "a straight line going through two points is the shortest distance between these two points" and "in a triangle, the addition of angles totals up to 180 degrees". The Euclidean subspace has been in use for a long time, after which non-Euclidean geometry (such as Riemannian [23] or Grassmannian [24] geometry) became popular since it is more natural and suitable for certain problems in mathematics and related areas such as computer vision and Artificial Intelligence. Moreover, the Euclidean space is not appropriate for working with non-linear manifolds due to the fact that Euclidean distance does not capture the intrinsic

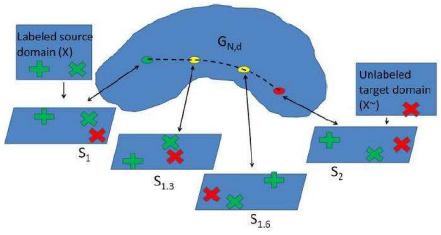


Figure 8: Illustration of sampling points between the subspaces of the source and target domains on the Grassmann manifold. In the figure, the Grassmann manifold is represented by $G_{N,d}$ which is basically the space of d-dimensional subspaces in $\mathbb{R}^N$ and S_1 and S_2 are two points on $G_{N,d}$ corresponding to the source and target domains respectively. The ones in between S_1 and S_2 can be considered as intermediate subspaces (*i.e.* intermediate points on the Grassman manifold) going from the source point to the target point. Figure taken from [25].

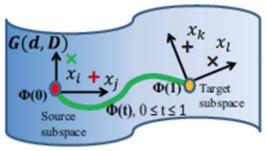


Figure 9: Geodesic flow kernel to model the gradual change from the source domain, $\Phi(0)$, to the target domain, $\Phi(1)$. $\Phi(t)$ gives the subspace at any point, *i.e.* $0 \leq t \leq 1$, along the geodesic. Figure taken from [27].

nonlinear geometric structure of data. In a non-flat (or curved) space, betwen any two points (i.e. a *geodesic*), there can be more than one shortest distance path. For example, there are several geodesics between the Earth's south and north poles. Unlike Euclidean distance, the geodesic distance takes into consideration the global nonlinear structure of data. This is highly useful for domain adaptation purposes in computer vision since visual data (raw image data in particular) often exist in a very high-dimensional space which actually lies in a much lower dimensional manifold (which corresponds to high level features and meaningful concepts in the image). This can be used to help with the domain adaptation.

Gopolan *et al*. [25] propose an unsupervised domain adaptation method that models the domain shift by gradual changes in the representation from the source to target domain. This is achieved by modelling the subspaces in the source and target domains and then generating intermediate subspaces between them as sampled points along the geodesic on the Grassmann manifold [26]. This is shown in Figure 8. Their approach requires tuning of many parameters including determining the finite number of subspaces to sample.

Gong *et al*. [27] present a system similar to [25]. They propose a method called "geodesic flow kernel" (illustrated in Figure 9) which is an improvement on [25] in that it eliminates the need to sample a finite number of subspaces and to tune as many parameters as [25] by "kernalizing" the approach of [25] and considering an infinite number of subspaces.

Mirrashed and Rastergari [28] approach unsupervised domain adaptation by learning a set of discriminative and invariant feature projections (into binary space) that models the class structures across source and target domains. Each of these projections is essentially a hyperplane in the feature space and a binary "attribute" is obtained by looking at which side of the hyperplane the data falls on. Using this set of projections (*i.e*. hyperplanes), the source dataset is projected into the binary space to get the binary attributes and the target classifier is obtained by training a classifier using the projected data. There is however no proper justification as to why the space should be binary in the first place and the proposed optimization algorithm is prone to local optima.

4.3. Domain Adaptation for Object Detection in Videos

Bose and Grimson [29] propose an unsupervised domain adaptation system for adapting a (baseline) detector trained on a far-field video (or a set of far-field videos) towards a different far-field video by a two-step self-training algorithm. In the first step, the baseline detector is used to score and label the unlabelled data (*i.e*. all sliding windows of frames in the video). A new classifier is then trained on the combination of the original data (from which the baseline detector was trained) and the most confidently scored data of the unlabelled data. In the second step, this newly trained detector is applied to the video and scene-specific features (*e.g*. silhouette height obtained by background subtraction) are extracted from the detections and a new classifier is trained with these features. There are a few limitations with this domain adaptation approach:

1. There is a need to determine the threshold for the "most confident" detections.

2. It is not known for sure whether the classifier obtained at the end of the first step is good enough. If it is not, then the second step will carry on the errors and may even make it worse. In other words, the second step is completely dependent on the outcome of the first step and has no chance of correcting any errors of the first step.

3. The final detector (at test time) is (still) dependent upon the results of background subtraction (in order to extract the scene-specific features). This can be a problem if the background subtraction is very noisy, especially for complex and cluttered scenes.

In addition, in the paper, it is not clear whether the performance improvement comes from the actual detector adaptation or from using a better (*i.e*. higher level) feature extraction mechanism.

A system that adapts a set of general part detectors to specific video scenes is proposed by Wu and Nevatia [30]. This is achieved by using a self-training framework where the "oracle"[1] is the (global) combination of the part detections; the

[1]The "oracle" is the verification process that selects which examples to include for each self-training iteration. It is often the most important component of a self-training algorithm. In order to maximise the efficiency and effectiveness of the self-training process and to minimise drifting, the oracle should be as independent as possible from the original (*i.e*. source) dataset and should offer complementary information to the information already contained in the source dataset.

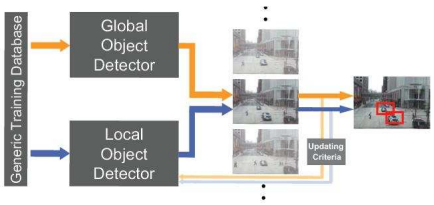


Figure 10: Detector adaptation approach of Kemhavi *et al*. [31] by combining the predictions of a fixed global detector and an online updated local detector. Figure taken from [31].

global shape model given by the configuration of parts provides an additional and complementary source of information compared to the (local) part detectors. The approach is limited to boosting-type classifiers and to object detection systems that explicitly model objects with parts.

A Multiple Kernel Learning based self-training algorithm is used by Kemhavi *et al*. [31] to tune a generic vehicle detector to a traffic intersection. Their adapted detector is a combination of two separate detectors: one is termed a "global detector" which is the detector trained on the generic dataset and fixed (*i.e*. no updates are performed), and the other is an online detector updated with a simple self-training approach: most confident positive and negative examples scored by the global detector are added in each round. This is shown in Figure 10. For adding negative examples, examples are added that are both confident and have high positional entropy relative to the positions (in the image plane) of the currently collected negative image patches. This is to prevent too many negative patches from the same background position from being added.

The process of using the global detector as an oracle in this way may not be very effective because the online classifier may never get better if the global detector (which is fixed) does not perform very well in the first place and additionally, the global detector does not provide any new complementary information to the online detector (since the online classifier is obtained from the global detector). Moreover, their method only applies to a particular type of classifier (*i.e*. Multiple Kernel Learning). Another potential problem is due to the final classifier being the combination of the global detector and the online classifier: there is a limit to the amount of adaptation the final classifier can undergo. For example, if the generic detector has a lot of false positives, it would still influence the final classifier to a large extent. And finally it is non-trivial to manually specify the best combination of the global detector and the online detector.

Wang *et al*. [32] propose a self-training algorithm to adapt a generic pedestrian detector to a specific scene. Their algorithm does not utilise background subtraction or (explicit) object tracking and it works as follows. Firstly the detector is applied on frames of the video with a high recall and low precision setting. Then a hierarchical k-means tree is constructed using the features of these detections. Thirdly, the most positive and negative detections are identified and they are encoded using the learnt tree to obtain binary codes and a classifier is trained on this binary feature space. This is the scene-specific detector

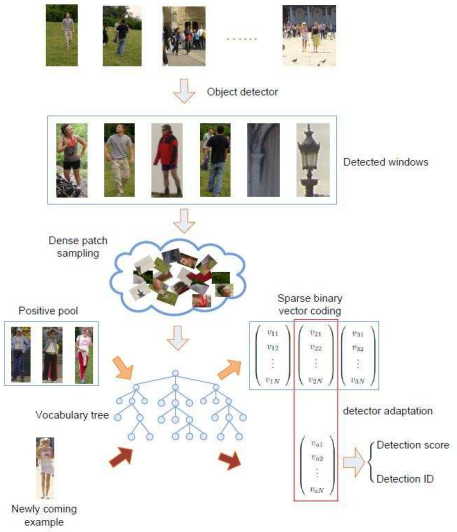


Figure 11: System overview of the algorithm of Wang *et al*. [32]. Figure taken from [32].

(illustrated in Figure 11). The performance is sensitive to setting and manual tuning of many parameters such as choosing a suitable low precision and high recall setting, thresholds for collecting confident positive and negative examples, the depth of the hierarchical k-means clustering and parameters for similarity measure for the binary features. It is uncertain whether the improvement of the scene-specific detector over the generic detector comes from the adaptation stage or the non-linear feature encoding stage (two unrelated steps). Furthermore, it is highly likely for the first step to fail to collect sufficient labelled data (due to collecting only the most confident positive and negative examples) to train a scene-specific detector that has good generalisation properties in the target domain.

Another self-training method is proposed by Sharma *et al*. [33] to adapt a generic detector to specific scenes for the task of pedestrian detection. The classifier used is Real Adaboost [34] with Multiple Instance Learning [35] loss function. The self-training approach works by applying the current detector to frames and then associating the detections into tracks. Then successfully tracked detections are added as new positive examples, and detections that do not belong to any of the tracks, are considered as new negative examples. For each detection to be added as a positive example, instead of directly adding the patch corresponding to the detection, the original patch and multiple patches surrounding the patch are treated as samples in a positive bag with the assumption that one of these patches contain the correctly localised positive example (as in the standard Multiple Instance Learning framework) . This is used to reduce the patch alignment errors commonly associated with collecting examples from detections.

Their approach however is limited to Real Adaboost classifiers and the datasets that they are evaluating their algorithms on are high resolution datasets and only one type of moving object (*i.e*. pedestrian) is present in the scene. The Multiple Instance Learning approach that they have adopted is not likely

to work well for low resolution videos such as far-field surveillance scenes.

Tang *et al*. [36] adapts a detector trained on an image dataset to a video based on a variation of iterative self-training which they term "self-paced domain adaptation". It works by adding easiest examples to the dataset first followed by increasingly more challenging ones. However, the self-paced domain adaptation technique is not much different from the traditional self-training approaches which seek to iteratively add the most confident detections in each round to *slowly* adapt the classifier to minimise the risk of drifting. For selecting examples to add in each round, instead of scoring individual detections, they score tracks in order to average out noise associated with individual detections. They assume that negative examples are known in the scene which means that their approach requires partial supervision.

To adapt a face detector in the form of a pre-trained cascade of classifiers to a new domain, Jain and Farfade [37] use a supervised domain adaptation algorithm. Their approach is essentially a type of self-training method where the oracle is a generative appearance model. They tested their algorithm by adapting a generic frontal face detector (such as the one available in the OpenCV library) to images containing baby faces. The approach however is limited to classifier cascades, requires a few hundreds of labelled annotation in the target domain and therefore is labour-intensive.

Sharma and Nevatia [38] present a self-training approach to adapt a pedestrian detector to video scenes. In order to collect samples for self-training, they apply the baseline detector and keep only the most confident detections. Then the detections are placed into tracks using appearance, size and position cues. After the samples are collected, the positive examples are divided into different subcategories by applying a pre-trained pose classifier. Then they train a random fern classifier [39] for each positive subcategory to increase the precision of the baseline detector.

From the evaluation in [38], it is not clear whether the detection improvement comes from the subcategory division and training nonlinear random fern classifiers or the actual adaptation algorithm itself. Moreover, the method requires a pose classifier for pedestrians to be trained and also involves non-trivial tuning of multiple parameters such as the thresholds for applying the detector in "high precision setting" for collecting samples during the adaptation stage and "high recall setting" at test time. Lastly, the adapted algorithm only reduces false positives and does not increase recall.

A co-training approach is adopted by Mirrashed *et al*. [40] to adapt vehicle detectors from multiple source domains to a target domain. Classifiers trained on different source domains iteratively train and improve each other by teaching, in each iteration, the most confident detections of one classifier to the other classifier(s). The algorithm makes use of Transfer Component Analysis [41] in order to reduce the effects of domain shifts between the datasets. As with other iterative self-training algorithms, the algorithm requires setting the threshold for selecting confident detections. Moreover, the system requires the use of multiple source domains which may not be feasible in many situations.

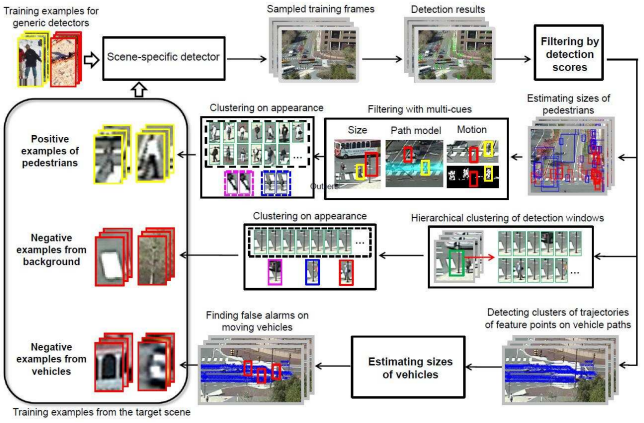


Figure 12: An iterative self-training technique of Wang and Wang [11]. In each iteration, positive and negative examples are collected by filtering with a variety of cues, added to the current dataset and a new classifier is trained. Figure taken from [11].

Shu *et al*. [42] propose a self-training approach to adapt a generic pedestrian to specific videos. Firstly, the generic detector is applied on frames and then the most confident detections are collected as positive examples. Negative examples are collected from the scene background. Then super-pixels are extracted and patches corresponding to the super-pixels are clustered to form a visual dictionary. This is used to encode the examples using a Bag of Words (BOW) approach. Then a classifier is trained using the encoded examples. Next, the classifier is applied on frames and this process repeats until convergence.

The approach also requires setting of several sensitive hyper-parameters such as the parameters of super-pixel generation, the number of clusters for building the dictionary and the confidence threshold for positive sample collection. Since the negative examples come only from the scene background, the algorithm may not work well for videos where there are multiple moving objects. The evaluation is performed only on quite straightforward datasets where there are only pedestrians moving about. Furthermore, the super-pixel extraction may not work well for videos where pedestrians are medium or small-sized. Most importantly, it is again not clear whether the adaptation performance actually comes from the adaptation algorithm or from using a better feature extraction mechanism than the baseline detector. This is important because if the baseline classifier uses the same feature extraction mechanism (*i.e*. super-pixel generation and BOW encoding) then it may be as good as the final classifier. If this is the case, then it would imply that the major part of the novelty is not in detector adaptation but in feature engineering.

The method proposed by Wang and Wang [11] iteratively improves a generic pedestrian detector by selecting new confident examples to add to the current dataset for retraining at every iteration. In order to collect examples for each self-training iteration, their oracle is a combination of vehicle and pedestrian paths, multiple different cues such as bounding box locations and sizes, background subtraction, thresholds, filters and hier-

archical clustering. To obtain vehicle and pedestrian paths, they use the method of [43] which discovers motion patterns in long-term videos using topic models such as Hierarchical Dirichlet Processes [44]. The motion patterns are discovered in a bottom-up manner by treating quantized optical flow velocity and position (in the image plane) as low-level features, small video clips as documents and then co-clustering using the topic models.

The approach requires quite extensive parameter setting and tuning such as deciding the length of a video segment (for topic modelling), setting the hyper-parameters for optimizing the topic model and determining various parameters for different filtering steps, clustering and background subtraction and thresholds for object sizes. There is also a need to manually label the discovered paths and an assumption that pedestrians and vehicle paths are not overlapped to a certain degree. Lastly, the number of iterations for self-training is also required to be set and there is a possibility of drifting if too many iterations are performed. The overview of their system is shown in Figure 12.

The method is extended in [10] by incorporating techniques such as reweighting the source data, confidence propagation and using the confidence when retraining rather than hard thresholding. Using a much simpler and a more efficient non-iterative algorithm, an improvement in state-of-the-art results on these datasets was proposed by [45].

Recently, methods based on deep learning have also been attempted for the purpose of domain adaptation [46, 47, 48, 49]. However, as most of them are using different datasets with various network configurations, architectures and parameter settings and tunings. Thus, it is not possible, at this point in time, to compare their approaches and the corresponding results in a fair and transparent manner and it is unclear how well deep learning really performs for use in domain adaptation. It is an open and interesting research question that could be addressed in the future.

4.4. Areas Related to Domain Adaptation

We now describe three areas of study somewhat related to domain adaptation for object detection in videos. Firstly, in Section 4.4.1, we highlight research on learning general moving object detectors in video. Secondly, in Section 4.4.2, we discuss semi-supervised learning of object detectors in video. And finally in Section 4.4.3, we end with weakly supervised learning of object detectors.

4.4.1. Learning moving object detectors

The most common way of detecting foreground objects in video is to use background subtraction followed by a grouping technique such as Connected Component Analysis [50]. For more information on different approaches to background subtraction, the reader is referred to various surveys [51, 52, 53].

In this subsection, we focus on approaches based on training (*i.e*. learning) *classifiers* to model and detect general foregrounds (*i.e*. significant objects) in the scene, often improving the results of traditional background subtraction approaches by utilising the *generalising* (and noise-reduction) power afforded by the classifier training stage.

Figure 13: Office corridor scene used in [54]. Figure taken from [54].

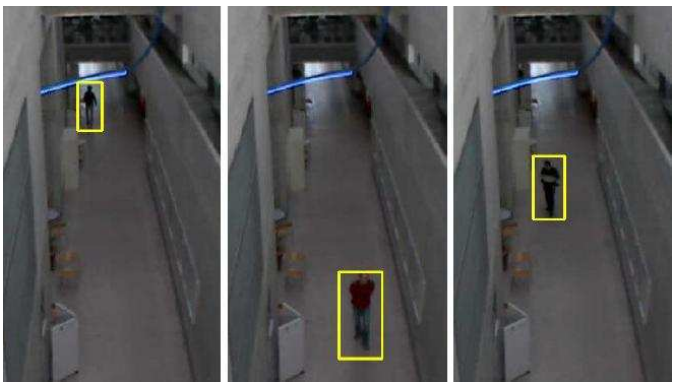


Figure 14: Indoor scene for pedestrian detection in [56]. Figure taken from [56].

The research problem tackled by these approaches is different from domain adaptation explored in this paper. Moreover, their goal is to "blindly" detect *any* foreground object in the scene as opposed to being *aware* of specific object classes and detecting them in the scene. However, we review these papers for the sake of completeness since some of these methods do use self-training-like algorithms.

Nair and Clark [54] propose an approach for online learning of a moving object detector for an office corridor scene. An online Winndow classifier is trained on features extracted from foreground blobs obtained by background subtraction if foreground blobs have the correct aspect ratio and size corresponding to pedestrians. They evaluate their approach only on indoor scenes (shown in Figure 13) where there is only one type of moving object (*i.e*. pedestrian) for which background subtraction already performs quite well due to the restricted environment, where there are no major problems such as background clutter, multiple categories of objects, large illumination variation and large cast shadows (which would change the aspect ratios and sizes of detected blobs) that would be common in a lot of outdoor surveillance type scenarios. The (intra-class variation of) background clutter of their indoor scene is quite small, making the dataset not particularly challenging. A similar system using online Adaboost is proposed by Roth *et al*. [55].

Grabner *et al*. [56] propose a "grid-based" pedestrian detection[2] system based on training one classifier for each image location (in the form of a pedestrian-sized window) and updating

[2] As will be explained later, their approach actually reduces to a foreground detection system in scenes where there is more than one category of moving objects. This is why we discuss their approach here.

them independently online based on a simple update heuristic. The update strategy works as follows: they fix the positive class (with a small number of pedestrian examples) for all the classifiers without any update and *always* update the negatives with the assumption that the probability of wrongly updating the negatives is very small. The method assumes that the intra-class variation of the negative class (*i.e*. non-pedestrian patches) at each image location is extremely small and takes advantage of this to simplify the complexity of each classifier. While this may be the case for some scenes, it is not true for many scenes especially those where there is more than one class of objects. In those types of scenes, many image locations would still have to handle large intra-class variations (given by the combination of intra-class variations of the background at that image location and other object categories that may occupy the image location at any time), rendering the original intention of simplifying the task of the classifier ineffective. There are a number of additional potential problems associated with the approach:

1. The positive class is fixed and never updated, which means that the system may never detect some pedestrians which are not well represented by the initial set of pedestrian examples.

2. The negative class is always updated, which means that the negative class of each classifier will be dominated by the background of the image location corresponding to the classifier. This means that other (*i.e*. non-pedestrian) classes of objects that occasionally move inside the image location would most likely be erroneously classified as "pedestrian" since the classifier will be quite certain that it does not belong to the negative class (dominated by the background).

3. If a pedestrian stays in a particular image location for a long time, all the pedestrian patches in this duration will be incorporated as "non-pedestrian" data and the resulting classifier at that image location would then learn to classify pedestrians as "non-pedestrians" with high probability (thereby decreasing the recall of the system).

4. Even though training one classifier per image location simplifies the task of each classifier, the combined complexity of all the classifiers is still much higher. And due to the fact that negative data are not *shared* among the individual classifiers, it can result in overfitting at the system level (even if there is no overfitting at the individual classifier level).

Because of these problems, the system may result in low recall and low precision simultaneously, especially in complex surveillance-type scenes with multiple object categories. Coincidentally, they evaluate their method only on relatively simple indoor scenes where there is only one class of moving objects as shown in Figure 14. In fact, rather than "pedestrian detection", the system is more similar to the traditional background subtraction and if applied to more complex scenes, it would not be much different than block-based background subtraction approaches such as [57]. A similar system is also proposed by Roth *et al*. [9].

Stalder *et al*. [58] extend [56] by updating both the positive and negative classes in each image location (*i.e*. for each classifier) and proposing different update strategies than [56]. The positive class for each image location is updated using the current patch if it is verified by a fixed generic detector (which is a global detector independent from the grid classifiers) or 3D context (*e.g*. assumption of a common ground plane). Negative class for each image location is updated by background images at that location obtained by a long-term generative (pixel-based) background subtraction algorithm.

Although the paper proposes more complex update heuristics than [56] for the classifiers, it also somewhat defeats the original purpose of having these grid-based classifiers, which is to make the task of each classifier simple and robust to drifting (at least for the positive class) by adopting fixed updating strategies. Compared to [56], their approach opens up the possibility of positive class drifting. Moreover, updating the negative class with the results of background subtraction introduces errors associated with most pixel-based and generative background subtraction methods. This problem is minimised in [56] by avoiding pixel-wise background modelling and instead, by modelling the large neighbourhood of pixels in a discriminative fashion. Therefore, in [58], the need for grid-based classifiers is no longer obvious. Furthermore, it also still shares a few limitations of [56]. And lastly, it requires the assumption and estimation of a single ground plane and 3D context which may not be readily available.

4.4.2. Semi-supervised learning for object detection

In this section, we briefly review work on semi-supervised learning of object detectors for videos. However, this work solves a different problem than domain adaptation (*i.e*. semi-supervised learning setting assumes that there is no source domain and some labelled data are always given in the target domain) but we include these here for completeness.

Levin *et al*. [59] propose a co-training [60] approach for semi-supervised learning of vehicle detectors in video. Given some labelled data in the target scene, firstly, a pair of car detectors is trained; one of the pairs is trained on data for whose feature extraction is performed on original images and for the other, background subtracted images instead of the original images are used. Then these two classifiers are used to teach and improve each other by "feeding" one the confident detections of the other and retraining the classifiers. They tested their methods on videos of vehicles on a highway captured by a surveillance camera. A similar co-training system is presented by Javed *et al*. [61] by using online boosting.

Rosenberg *et al*. [62] use iterative self-training for semi-supervised learning of an eye detector. For the "oracle" (*i.e*. for selecting which examples to include for each iteration of self-training), instead of using the detector's own confidence, the system uses nearest neighbour scores of all examples in the current dataset to the detection, in an attempt to make the oracle independent from the detection. However, the oracle is still

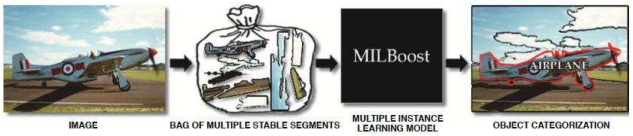


Figure 15: An image is represented by a *bag* of multiple stable segments which is obtained by collecting the outputs of different segmentation algorithms and various segmentation parameters with the assumption that one of the segmentations in the bag would correctly correspond to the aeroplane. Then by looking at multiple such bags corresponding to training images where each image contains an aeroplane, the aeroplane category can be inferred and segmented in each of the images. Figure taken from [64].

not independent because it is derived from the same dataset that the detector was trained from. Their approach can also be seen a type of co-training where two classifiers have two different classifier types (*i.e*. inductive biases) and one of the classifiers is fixed.

Ali *et al*. [63] propose an iterative self-training algorithm based on Adaboost for semi-supervised learning of a pedestrian detector in a video, given sparsely annotated video (*i.e*. a small subset of all the frames in the video are labelled). Examples to include for each iteration of the self-training is determined by track smoothness. The method is however limited to Adaboost and not applicable to other types of classifiers.

4.4.3. Weakly supervised learning for object detection

Galleguillos *et al*. [64] propose a weakly supervised approach to learn object detectors given weakly labelled images. In their case, weakly labelled images are considered as images containing the desired objects but the exact locations and spatial extent of those objects are not specified.

However, the method does require the objects to be spatially occupying the major portion of the images for their algorithm to work well. To our knowledge, they are the first to use the idea of "multiple stable segmentations" and Multiple Instance Learning (MIL) for the purpose of training object detectors using weakly labelled images.

Multiple stable segmentations is an idea that in any image containing an object of interest, an ensemble or bag of segmentations obtained by multiple segmentation algorithms and different segmentations parameters will most likely result in the object being *correctly* segmented in at least one of these segmentations in the ensemble. Each image containing an object can therefore be associated with a bag of segmentations from which one of them corresponding to the desired object. This is a much better and useful prior information than not having any information about the object in the image. Since an image containing an object can be represented with a bag (of possible objects), MIL can be used to learn the most consistent object category by optimizing across multiple such images and corresponding bags. This is illustrated in Figure 15.

Weber *et al*. [65] propose another weakly supervised training approach to learn human face models and models of rear views of cars. Again, similar to [64], their method assumes that each object occupies the major portion of the corresponding training image. They represent an object as a constellation

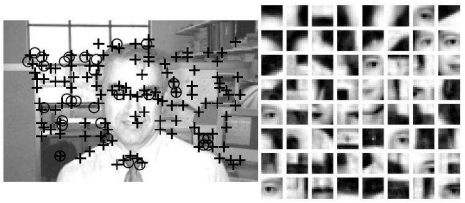


Figure 16: On the left is an example of interest point detection on an image containing a face. The right picture shows a set of distinctive parts discovered by clustering the patches corresponding to interest points across multiple training images containing faces. Figure taken from [65].

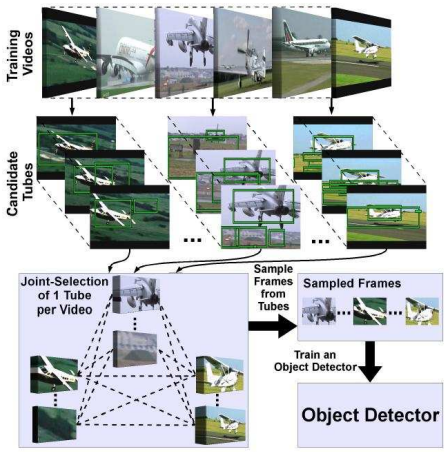


Figure 17: Overview of the weakly supervised learning approach by Prest *et al*. [69]. Figure taken from [69].

of parts where parts are detected by an interest point detector. Distinctive parts are discovered by clustering the detected parts (represented by features extracted from regions centred at the interest points) in the training images. This is shown in Figure 16. Then object classes are learnt by searching for parts and geometry of parts that are consistent across the training images.

The method requires high resolution images (for reliable part detection) and is dependent upon the interest point detector to consistently and correctly fire on actual part-like locations in images. Moreover, if the background clutter is high, the system may not correctly learn the desired models. Recently, methods such as [66] have also been been proposed to deal with situations where there are multiple objects of interest and significant background clutter in each image.

Blaschko *et al*. [67] and Pandey and Lazebnik [68] propose latent-SVM-based weakly supervised training algorithms where the bounding boxes of objects are treated as latent variables to be inferred during training. The general problem, however, with these approaches is that the resulting optimization function is very prone to get stuck in bad local optima unless there is a good initialisation. Although they propose some heuristics for initialising such models, they are usually application and object-specific. Moreover, the training algorithm can also be very computationally expensive.

Prest *et al*. [69] propose a weakly supervised learning of object detectors from short YouTube video clips where each video

clip is assumed to contain an object of interest moving about. A diagram illustrating the overview of their approach is shown in Figure 17.

Their method first identifies candidate spatio-temporal tubes from which at least one of them is very likely to contain a moving object of interest in each video clip. Then from many sets of these candidate spatio-temporal cubes from multiple video clips, a consistent class of spatio-temporal cubes is found by jointly considering all the spatio-temporal candidate cubes across all the training video clips and minimising an objective function.

Similar to many other weakly learning approaches, their approach also has an implicit assumption that objects of interest occupy the majority of the spatio-temporal volume in the video clips and the optimization algorithm can get stuck in bad local optima without a suitable initialisation which is non-trivial.

5. Discussion

In this section, we recap, put into context and compare the relevant representative papers that have been discussed in detail in Section 4.

One of the most relevant works for domain adaptation for images is by Gopalan *et al*. [25] who propose building intermediate representations between source and target domains by using geodesic flows. However, their approach requires sampling a finite number of subspaces and tuning many parameters such as the number of intermediate representations. Gong *et al*. [27] improves on [25] by giving a kernel version of [25]. However both [25, 27] are dealing only with domain adaptation for image classification as opposed to domain adaptation for object detection. Moreover, their approaches, unlike [66], do not learn deep representations required for manifolds that are highly non-linear.

As can be observed in Section 4.3, the overwhelming majority of the state-of-the-art research for domain adaptation of object detectors in videos use self-training in one form or another [29, 30, 31, 32, 33, 36, 37, 38, 40, 42, 11, 10]. In order to adapt a generic pedestrian detector to a specific scene, a typical system would run the generic detector on some frames in a video, then score each detection using some heuristics and afterwards, add the most confident positive and negative detections to the original dataset for retraining. This process is repeated over multiple iterations. Each of these approaches suffers from a subset of the following problems:

1. The need to manually determine and set thresholds for "the most confident" detections, "the least confident" detections, low precision and high recall settings and so on.

2. Many of them only work with specific types of classifier (such as Adaboost, cascaded classifiers and Multiple Instance Learning).

3. Most of them need setting of the number of iterations for the iterative self-training.

4. Many of them are prone to drifting since wrongly labelled examples in one iteration could make the detector become progressively worse in the following iterations.

5. Most of them require the presence of the original dataset for retraining. This is expensive especially for large datasets and many a time, we may only have a generic *detector* (which can be a classifier of any type) but not the generic *dataset* itself (due to copyright reasons, *etc*.).

6. Most approaches have several sensitive parameters to set and tune. And these parameters change for different videos (or scenes), many of them cannot be set automatically (for unsupervised domain adaptation) and for some, it is non-trivial to tune them automatically without extensive and very expensive cross validation.

7. Many of the approaches do not work well with low-resolution far-field surveillance videos.

8. Some of them require labelled data (*i.e*. supervision) in the target domain, *i.e*. they are supervised domain adaptation approaches.

We now go through each of the representative related works discussed Section 4.3 and briefly compare the papers.

Bose and Grimson[29] evaluate their approach on far-field surveillance scenes. However, most of the improvement of their detector adaptation comes from using a different and better feature extraction *at* test time. In contrast, in algorithms proposed in works such as [45, 10, 11], the majority of the benefit of domain adaptation is derived from the systematic and effective collection of scene-specific positive and negative examples. Therefore, unlike [29], approaches proposed in [45, 10, 11] can still improve performance further by extracting new and better features specific to the scene after collecting the scene-specific positive and negative data.

The approach by Wu and Nevatia [30] only works with part-based detectors, so it is not really suitable for the majority of holistic object detectors that we focus on in this paper.

The Multiple Kernel Learning approach of Kemhavi *et al*. [31] is expensive at test time as opposed to algorithms presented in [45, 10, 11] which can generate a linear classifier (or any type of classifier) for test time.

Wang *et al*. [32] and Sharma *et al*. [33] deal with domain adaptation for pedestrian detection, however their approach is not likely to work well for the low resolution videos. The *supervised* domain adaptation approach of Jain and Farfade [37] using cascade classifiers solves a different problem (*i.e*. (supervised domain adaptation) which is much easier than *unsupervised* domain adaptation. The paper by Mirrashed *et al*. [40] requires multiple (*i.e*. at least two) source domains whereas most of the works reviewed in this paper assume that only one source domain is available. The approach of Shu *et al*. [42] may work poorly for videos with small pedestrians.

Compared to training object detectors using strong supervision, the literature concerning weakly supervised training is limited. In the existing approaches, supervision is given in the

form of image-level labels where the exact location and spatial extent of objects of interest are considered unknown and treated as latent variables to be inferred from data during training.

One of the ways of solving this problem is by formulating it as Multiple Instance Learning (MIL) [35, 70] in which supervision labels are given at the *bag* level rather than at the instance level. Each positive bag is assumed to contain at least one positive instance and each negative bag is assumed to contain all negative instances. In order to generate positive bags and because the space of all possible object locations and sizes is too large to be tractable during training, many existing approaches use an ensemble of low-level segmentations to generate numerous candidate regions with the assumption that at least one of them contains the desired object [64, 71]. The output of such a system, however, depends heavily on the results of segmentation.

Furthermore, most existing approaches work with datasets where an object occupies a large central portion of each image in most of the training images [72, 73, 71, 64]. This is in contrast to [66] which is dealing with far-field videos where there are often multiple objects of varying sizes in each frame and each object occupies only a tiny portion of a frame. Moreover, [66] can work with low-resolution objects that do not allow sophisticated part-based modelling and discovery.

Deselaers *et al*. [74] propose an iterative algorithm to learn object classes from weakly supervised images using a conditional random field that progressively adapts to the new classes. Chum and Zisserman [72] give an algorithm that locates image regions corresponding to object classes of a set of training images by optimizing an objective function that computes similarity between pairs of images.

Considering classifier parameters and subwindows of objects jointly as latent variables in an SVM classification objective function, Nguyen *et al*. [73] optimize the function to infer the variables. Weakly supervised learning is tackled as a structured output learning framework in [67].

Most of the aforementioned approaches deal only with images and do not make use of information that can be exploited in surveillance-type videos.

Recently, Prest *et al*. [69] propose a weakly supervised learning approach for YouTube video clips. Their approach, which is essentially an extension of [74] to video, solves a fundamentally different problem from most of other papers in that they assume that small independent video clips are the training data and each video clip contains the desired object class in a large proportion of the spatio-temporal volume.

We now make some practical recommendations on which methods should be preferred for certain situations.

If the target dataset (*i.e*. domain) is a video captured with a static camera, it is best to use the iterative self-training algorithms proposed in [10, 11, 45] because it makes maximum use of cues available in video, resulting in the highest domain adaptation accuracy. Moreover, not only it is reasonably efficient and fast during training (*i.e*. during domain adaptation), it is also very fast at test time since there is no need to perform expensive feature projection (as required by the domain adaptation algorithm using feature projection approaches in [25, 27, 28, 47]). In addition to this, if there is a generic detector but if the corresponding generic dataset is not available, the non-iterative self-training method in [45] should be used since the algorithms in [10, 11] requires the generic dataset to be present.

If smooth spatio-temporal constraints cannot be reliably exploited in the target domain (either due to the video camera recording at very low frame rates, due to the presence of sufficiently large camera movements or due to the fact that the target domain is a set of static image collections with no temporal connections), we would recommend using the feature learning and projection approaches (such as in [46, 47, 48, 49]). However, with these approaches, if faster pedestrian detection is desired, we would recommend that during test time, rather than exhaustive sliding window detection, some other methods (such as 3D or ground plane information) should be used to limit the the number of sliding windows that need to be evaluated.

Finally, in a situation where neither the generic dataset nor the generic detector is present, domain adaptation is then not possible and weakly-supervised learning approaches (*e.g*. [69, 74, 66]) should be used.

6. Conclusion & Future Directions

Due to the need for high performance in the automated analysis of the ever increasing amount of visual data, domain adaptation has become popular in recent years in the fields of computer vision and machine intelligence. In this paper, we survey, review and analyse the most relevant works for domain adaptation in the context of pedestrian detection in image and video data. In order to provide readers with the necessary background, a brief tutorial on transfer learning and domain adaptation is also presented. Furthermore, in order to benefit practitioners in real life scenarios, we make recommendations on which domain adaptation methods should be preferred in specific situations.

In this field, there are a number of promising research directions that can be identified. Firstly, it would be beneficial to combine the two main streams of domain adaptation techniques, which are iterative self-training and learning common representations across the source and target datasets. By integrating these two categories of methods in the future, we expect that their advantages can be brought together and some of their disadvantages can be eliminated, resulting in even higher domain adaptation performance.

Secondly, it can be observed that most of the existing domain adaptation systems in the literature require manual tuning of sensitive hyper-parameters. This is especially more true with the unsupervised domain adaptation methods. It is an open research area to investigate how to perform "hands-off" fully-automated domain adaptation that can simply be deployed without the need for further intervention or assistance from human operators. For supervised domain adaptation, it would be interest to characterize the amount and nature of supervision required in the target dataset.

Thirdly, a general method that can detect concept drifting would be highly beneficial not only for domain adaptation, but also for related areas such as semi-supervised learning.

And finally, a promising research area would be to integrate domain adaptation with active learning, semi-supervised learning and weakly supervised learning, with the goal of minimising the cost (in terms of time, money, labour, *etc*.) involved in training accurate object detectors. This will have wide-ranging benefits in the area of Artificial Intelligence since objects form the basic building blocks in reasoning about the visual world.

Acknowledgement

We duly acknowledge the financial support from the Fully Funded International Research Scholarship (FIRS), University of Leeds, UK. This work was carried out while the first author was at University of Leeds.

References

[1] A. Andreopoulos, J. K. Tsotsos, 50 years of object recognition: Directions forward, Computer Vision and Image Understanding (2013) 827–891.

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

[3] M. Enzweiler, D. M. Gavrila, Monocular pedestrian detection: Survey and experiments, IEEE Transactions on Pattern Analysis and Machine Intelligence 31 (12) (2009) 2179–2195.

[4] D. Geronimo, A. M. Lopez, A. D. Sappa, T. Graf, Survey of pedestrian detection for advanced driver assistance systems, IEEE Transactions on Pattern Analysis and Machine Intelligence 32 (7) (2010) 1239–1258.

[5] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2005, pp. 886–893.

[6] R. B. Girshick, P. F. Felzenszwalb, D. A. Mcallester, Object detection with grammar models, in: Proceedings of Advances in Neural Information Processing Systems (NIPS), 2011, pp. 442–450.

[7] P. Dollár, C. Wojek, B. Schiele, P. Perona, Pedestrian detection: A benchmark, in: Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2009, pp. 304–311.

[8] A. Torralba, A. A. Efros, Unbiased look at dataset bias, in: Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2011, pp. 1521–1528.

[9] P. M. Roth, S. Sternig, H. Grabner, H. Bischof, Classifier grids for robust adaptive object detection, in: Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2009, pp. 2727–2734.

[10] M. Wang, W. Li, X. Wang, Transferring a generic pedestrian detector towards specific scenes, in: Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2012, pp. 3274–3281.

[11] M. Wang, X. Wang, Automatic adaptation of a generic pedestrian detector to a specific traffic scene, in: Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2011, pp. 3401–3408.

[12] J. Ferryman, A. Shahrokni, PETS 2009: Dataset and challenge, in: IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS-Winter), 2009, pp. 1–6.

[13] S. J. Pan, Q. Yang, A survey on transfer learning, IEEE Transactions on Knowledge and Data Engineering 22 (10) (2010) 1345–1359.

[14] M. Everingham, L. J. V. Gool, C. K. I. Williams, J. M. Winn, A. Zisserman, The Pascal Visual Object Classes (VOC) challenge, International Journal of Computer Vision 88 (2) (2010) 303–338.

[15] G. B. Huang, M. Ramesh, T. Berg, E. Learned-Miller, Labeled faces in the wild: A database for studying face recognition in unconstrained environments, Tech. Rep. 07-49, University of Massachusetts, Amherst (October 2007).

[16] K.-C. Lee, J. Ho, D. Kriegman, Acquiring linear subspaces for face recognition under variable lighting, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (5) (2005) 684–698.

[17] J. Blitzer, R. T. McDonald, F. Pereira, Domain adaptation with Structural Correspondence Learning, in: Conference on Empirical Methods in Natural Language Processing (EMNLP), 2006, pp. 120–128.

[18] D. McClosky, E. Charniak, M. Johnson, Reranking and self-training for parser adaptation, in: Annual Meeting of the Association for Computational Linguistics (ACL), Association for Computational Linguistics, 2006, pp. 337–344.

[19] B. Roark, M. Bacchiani, Supervised and unsupervised PCFG adaptation to novel domains, in: North American Chapter of the Association for Computational Linguistics: Human Language Technologies (HLT-NAACL), 2003, pp. 126–133.

[20] R. Hwa, Supervised grammar induction using training data with limited constituent information, in: Annual Meeting of the Association for Computational Linguistics (ACL), 1999, pp. 73–79.

[21] K. Saenko, B. Kulis, M. Fritz, T. Darrell, Adapting visual category models to new domains, in: Proceedings of the European Conference on Computer Vision (ECCV), 2010, pp. 213–226.

[22] B. Kulis, K. Saenko, T. Darrell, What you saw is not what you get: Domain adaptation using asymmetric kernel transforms, in: Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2011, pp. 1785–1792.

[23] L. P. Eisenhart, Riemannian geometry, Princeton university press, 1997.

[24] P. Turaga, A. Veeraraghavan, R. Chellappa, Statistical analysis on stiefel and grassmann manifolds with applications in computer vision, in: Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, IEEE, 2008, pp. 1–8.

[25] R. Gopalan, R. Li, R. Chellappa, Domain adaptation for object recognition: An unsupervised approach, in: Proceedings of the International Conference on Computer Vision (ICCV), 2011, pp. 999–1006.

[26] J. W. Milnor, Characteristic classes, Princeton University Press, 1974.

[27] B. Gong, Y. Shi, F. Sha, K. Grauman, Geodesic flow kernel for unsupervised domain adaptation, in: Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2012, pp. 2066–2073.

[28] F. Mirrashed, M. Rastegari, Domain adaptive classification, in: Proceedings of the International Conference on Computer Vision (ICCV), 2013, pp. 2608–2615.

[29] B. Bose, W. E. L. Grimson, Improving object classification in far-field video, in: Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2004, pp. 181–188.

[30] B. Wu, R. Nevatia, Improving part based object detection by unsupervised, online boosting, in: Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2007, pp. 1–8.

[31] A. Kembhavi, B. Siddiquie, R. Miezianko, S. McCloskey, L. S. Davis, Incremental multiple kernel learning for object recognition, in: Proceedings of the International Conference on Computer Vision (ICCV), 2009, pp. 638–645.

[32] X. Wang, G. Hua, T. X. Han, Detection by detections: Non-parametric detector adaptation for a video, in: Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2012, pp. 350–357.

[33] P. Sharma, C. Huang, R. Nevatia, Unsupervised incremental learning for improved object detection in a video, in: Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2012, pp. 3298–3305.

[34] J. Friedman, T. Hastie, R. Tibshirani, Additive logistic regression: a statistical view of Boosting, Annals of Statistics 28 (1998) 337–407.

[35] T. G. Dietterich, R. H. Lathrop, T. Lozano-Pérez, Solving the multiple instance problem with axis-parallel rectangles, Artificial Intelligence 89 (1) (1997) 31–71.

[36] L. F.-F. D. K. Kevin Tang, Vignesh Ramanathan, Shifting weights: Adapting object detectors from image to video, in: Proceedings of Advances in Neural Information Processing Systems (NIPS), 2012, pp. 638–646.

[37] V. Jain, S. S. Farfade, Adapting classification cascades to new domains, in: Proceedings of the International Conference on Computer Vision (ICCV), 2013, pp. 105–112.

[38] P. Sharma, R. Nevatia, Efficient detector adaptation for object detection in a video, in: Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2013, pp. 3254–3261.

[39] M. Özuysal, M. Calonder, V. Lepetit, P. Fua, Fast keypoint recognition using random ferns, IEEE Transactions on Pattern Analysis and Machine Intelligence 32 (3) (2010) 448–461.

[40] F. Mirrashed, V. I. Morariu, L. S. Davis, Sampling for unsupervised do-

main adaptive object detection, in: IEEE International Conference on Image Processing (ICIP), 2013, pp. 3288–3292.
[41] S. J. Pan, I. W. Tsang, J. T. Kwok, Q. Yang, Domain adaptation via transfer component analysis, in: International Joint Conference on Artificial Intelligence (IJCAI), 2009, pp. 1187–1192.
[42] G. Shu, A. Dehghan, M. Shah, Improving an object detector and extracting regions using superpixels, in: Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2013, pp. 3721–3727.
[43] X. Wang, X. Ma, W. E. L. Grimson, Unsupervised activity perception in crowded and complicated scenes using hierarchical bayesian models, IEEE Transactions on Pattern Analysis and Machine Intelligence 31 (3) (2009) 539–555.
[44] Y. W. Teh, M. I. Jordan, M. J. Beal, D. M. Blei, Hierarchical Dirichlet Processes, Journal of the American Statistical Association 101 (476) (2006) 1566–1581.
[45] K. K. Htike, D. Hogg, Efficient non-iterative domain adaptation of pedestrian detectors to video scenes, in: Pattern Recognition (ICPR), 2014 22nd International Conference on, IEEE, 2014, pp. 654–659.
[46] X. Glorot, A. Bordes, Y. Bengio, Domain adaptation for large-scale sentiment classification: A deep learning approach, in: Proceedings of the 28th International Conference on Machine Learning (ICML-11), 2011, pp. 513–520.
[47] K. K. Htike, D. Hogg, Unsupervised detector adaptation by joint dataset feature learning, in: Computer Vision and Graphics, Springer, 2014, pp. 270–277.
[48] S. Chopra, S. Balakrishnan, R. Gopalan, Dlid: Deep learning for domain adaptation by interpolating between domains, in: ICML workshop on challenges in representation learning, Vol. 2, 2013, p. 5.
[49] Y. Bengio, Deep learning of representations for unsupervised and transfer learning, Unsupervised and Transfer Learning Challenges in Machine Learning 7 (2012) 19.
[50] H. Samet, M. Tamminen, Efficient component labeling of images of arbitrary dimension represented by linear bintrees, IEEE Transactions on Pattern Analysis and Machine Intelligence 10 (4) (1988) 579–586.
[51] M. Piccardi, Background subtraction techniques: a review, in: IEEE International Conference on Systems, Man and Cybernetics (SMC), 2004, pp. 3099–3104.
[52] S. Brutzer, B. Höferlin, G. Heidemann, Evaluation of background subtraction techniques for video surveillance, in: Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2011, pp. 1937–1944.
[53] S.-C. S. Cheung, C. Kamath, Robust techniques for background subtraction in urban traffic video, in: Proceedings of SPIE, Vol. 5308, 2004, pp. 881–892.
[54] V. Nair, J. J. Clark, An unsupervised, online learning framework for moving object detection, in: Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2004, pp. 317–324.
[55] P. M. Roth, H. Grabner, D. Skocaj, H. Bischol, A. Leonardis, On-line conservative learning for person detection, in: Proc. IEEE Intern. Workshop on Performance Evaluation of Tracking and Surveillance (PETS), 2005, pp. 223–230.
[56] H. Grabner, P. M. Roth, H. Bischof, Is pedestrian detection really a hard task?, in: Proc. IEEE Intern. Workshop on Performance Evaluation of Tracking and Surveillance (PETS), 2007, pp. 1–8.
[57] M. Heikkilä, M. Pietikäinen, A texture-based method for modeling the background and detecting moving objects, IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (4) (2006) 657–662.
[58] S. Stalder, H. Grabner, L. Gool, Exploring context to learn scene specific object detectors, in: Proc. IEEE Intern. Workshop on Performance Evaluation of Tracking and Surveillance (PETS), 2009, pp. 63–70.
[59] A. Levin, P. A. Viola, Y. Freund, Unsupervised improvement of visual detectors using co-training, in: Proceedings of the International Conference on Computer Vision (ICCV), 2003, pp. 626–633.
[60] A. Blum, T. Mitchell, Combining labeled and unlabeled data with co-training, in: The Annual Conference on Computational Learning Theory (COLT), ACM, 1998, pp. 92–100.
[61] O. Javed, S. Ali, M. Shah, Online detection and classification of moving objects using progressively improving detectors, in: Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2005, pp. 696–701.
[62] C. Rosenberg, M. Hebert, H. Schneiderman, Semi-supervised self-training of object detection models, in: IEEE Workshop on Applications of Computer Vision (WACV), 2005, pp. 29–36.
[63] K. Ali, D. Hasler, F. Fleuret, Flowboost - appearance learning from sparsely annotated video, in: Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2011, pp. 1433–1440.
[64] C. Galleguillos, B. Babenko, A. Rabinovich, S. J. Belongie, Weakly supervised object localization with stable segmentations, in: Proceedings of the European Conference on Computer Vision (ECCV), 2008, pp. 193–207.
[65] M. Weber, M. Welling, P. Perona, Unsupervised learning of models for recognition, in: Proceedings of the European Conference on Computer Vision (ECCV), 2000, pp. 18–32.
[66] K. K. Htike, D. Hogg, Weakly supervised pedestrian detector training by unsupervised prior learning and cue fusion in videos, in: Image Processing (ICIP), 2014 IEEE International Conference on, IEEE, 2014, pp. 2338–2342.
[67] M. Blaschko, A. Vedaldi, A. Zisserman, Simultaneous object detection and ranking with weak supervision, in: Proceedings of Advances in Neural Information Processing Systems (NIPS), 2010, pp. 235–243.
[68] M. Pandey, S. Lazebnik, Scene recognition and weakly supervised object localization with deformable part-based models, in: Proceedings of the International Conference on Computer Vision (ICCV), IEEE, 2011, pp. 1307–1314.
[69] A. Prest, C. Leistner, J. Civera, C. Schmid, V. Ferrari, Learning object class detectors from weakly annotated video, in: Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2012, pp. 3282–3289.
[70] S. Andrews, I. Tsochantaridis, T. Hofmann, Support Vector Machines for Multiple-instance Learning, in: Proceedings of Advances in Neural Information Processing Systems (NIPS), 2002, pp. 561–568.
[71] Y. Chen, J. Z. Wang, Image categorization by learning and reasoning with regions, The Journal of Machine Learning Research 5 (2004) 913–939.
[72] O. Chum, A. Zisserman, An exemplar model for learning object classes, in: Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), IEEE, 2007, pp. 1–8.
[73] M. H. Nguyen, L. Torresani, F. de la Torre, C. Rother, Weakly supervised discriminative localization and classification: a joint learning process, in: Proceedings of the International Conference on Computer Vision (ICCV), 2009, pp. 1925–1932.
[74] T. Deselaers, B. Alexe, V. Ferrari, Localizing objects while learning their appearance, in: Proceedings of the European Conference on Computer Vision (ECCV), 2010, pp. 452–466.