Weakly Supervised Object Detector Learning with Model Drift Detection

Parthipan Siva and Tao Xiang
School of Electronic Engineering and Computer Science
Queen Mary University of London, London E1 4NS, UK
{psiva,txiang}@eecs.qmul.ac.uk

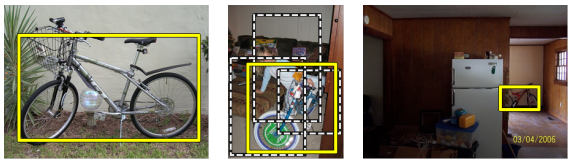
Abstract

A conventional approach to learning object detectors uses fully supervised learning techniques which assumes that a training image set with manual annotation of object bounding boxes are provided. The manual annotation of objects in large image sets is tedious and unreliable. Therefore, a weakly supervised learning approach is desirable, where the training set needs only binary labels regarding whether an image contains the target object class. In the weakly supervised approach a detector is used to iteratively annotate the training set and learn the object model. We present a novel weakly supervised learning framework for learning an object detector. Our framework incorporates a new initial annotation model to start the iterative learning of a detector and a model drift detection method that is able to detect and stop the iterative learning when the detector starts to drift away from the objects of interest. We demonstrate the effectiveness of our approach on the challenging PASCAL 2007 dataset.

1. Introduction

Object detectors [10] learn to localise objects of interest in images. Most existing methods for training object detectors take a supervised learning (SL) approach where a positive training set is manually annotated with a bounding box around each object of interest. Manual annotation of object bounding boxes in a large training set is tedious and unreliable. It is much easier and less ambiguous to only assign binary labels regarding whether an image contains any instances of a target object class. The problem of training an object detection model thus becomes a weakly supervised learning (WSL) problem.

Training an object detector using WSL attempts to address two sub-problems simultaneously: localising the objects of interest in each positive training image (automated annotation), and training a detector based on the automated annotation results (detector learning). This leads to a chicken-and-egg scenario. To localise objects accurately, one needs a strong detector that captures well the visual appearance of the target object class. However, that strong detector can only be obtained by learning from accurately localised object examples. In practice, one has to start somewhere, so an initial annotation is first obtained independently from the detector; a detector is then trained with the annotation and used as an annotator itself to refine the annotation, which in turn leads to the training of an improved detector. In essence, WSL for object detection is similar to self-training [4] although the training data is not completely unlabelled. It thus suffers from the model drift problem, that is, when the initial annotation is inaccurate, or wrong annotations are introduced in the iterative learning process, the model can drift away quickly. A similar problem is faced in detection-based object tracking [15].



(a) (b) (c)

Figure 1. Simultaneous localization and detector training is made challenging by changes in viewpoint, scale and background clutter.

In addition to the model drift problem intrinsic to WSL, there are other challenges unique to the object detection problem. Early attempts at WSL of object detector [11, 5, 12, 13, 20, 3] focus on images where the objects of interest, typically of a single viewpoint, dominate the scene with little background clutter (Fig. 1(a)). More recent efforts [19, 6] tackle the problem under a more realistic setting where objects of large intra-class variation are captured with a cluttered background (see Fig. 1(b)& 1(c)). In this case, the WSL problem can be re-cast as a multi-instance learning (MIL) problem [8]. In a MIL framework, one has a set of positive bags and negative bags. Each bag contains a set of instances such that a positive bag has at least one positive instance and a negative bag has no positive instances.

2011 IEEE International Conference on Computer Vision
978-1-4577-1102-2/11/$26.00

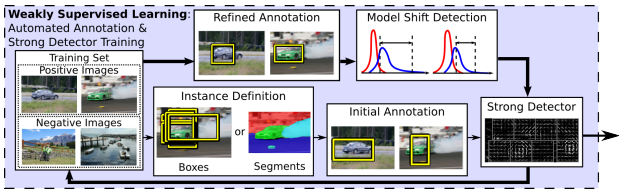


Figure 2. A schema of the proposed approach.

One aims to both localise the positive instances in those positive bags (annotation) and learn a positive instance classifier (detector learning). This brings about additional challenges: 1) instances cannot be clearly defined. They can be image windows, in which case there will be a large number of highly overlapping instances, either positive or negative within one positive bag (see Fig. 1(b)), or they can be image segments, in which case the object of interest might not be correctly segmented. 2) Due to the large intra-class variation, in particular the multi-modal distribution of object appearance caused by variable viewpoints (see Fig. 1), positive instance can look more similar to negative instances than to each other. Due to these challenges, both obtaining a good initial annotation and preventing model drift become even more critical.

The existing MIL methods are unsuitable for addressing these challenges. Specifically, they have two main limitations: 1) No measure is taken to prevent model drift. Using an existing method, the iterative learning process terminates when the automated annotation result converges. However, by then the model may have drifted to capture something other than the object of interest, e.g. commonly appearing background in positive images. 2) When selecting positive instances for detector learning, they do not explicitly take into account the multi-modal distribution of object appearance, most likely caused by variation in viewpoint.

To overcome these limitations, a new framework is proposed (Fig. 2). This framework has two novel components. The first component is a new annotation initialisation model which aims to give the iterative self-training process the best possible start. This model explicitly accounts for the multi-modal object appearance distribution in a real-world object detection task. The second novel component is a model drift detection method. Detecting model drift using weakly labelled data is extremely difficult because one does not know what the objects of interest look like in the first place without the exact object locations in positive images; one thus cannot directly measure whether the model has drifted to model something else. This is probably the reason why it has never been attempted before. In this work we propose to use a High Precision Range (HPR) to measure the change over consecutive iterations of the difference between the detection score distribution on the positive instances, selected by the current model, and that the negative instances in the negative bags. This simple intuitive measure is shown to be highly effective on detecting model drift in our experiments.

Related Works. A good initial annotation is critical for the iterative learning of an object detector. In theory, any MIL technique can be used for the initial annotation. A MIL method can select a set of positive instances based on two criteria: maximising inter-class distances (distance between the positive and negative instances/bags) and minimising intra-class distances (distances between positive instances/bags). These criteria can be applied to select one instance at a time (local selection) or all instances simultaneously (global selection). Ideally, a MIL method should exploit both criteria globally. However, most existing methods fall into two categories: global-Inter [2] or local-Intra-Inter [18]. Developing a global-Intra-Inter method is non-trivial and has only been attempted recently in [16] and [7]. However, both involve learning of complex graphical model and are considered to be too expensive for an initial annotation model. In this paper, a novel global-Intra-Inter initial annotation model is proposed which explicitly accounts for the multi-modal distribution of positive instances. It is thus well-suited for the task of automated annotation of object location in positive images.

Early works on applying MIL for solving vision problems have been focused on image categorisation [11, 5, 12, 13]. The most relevant works to this paper are those in [19] and [6], both of which try to solve the weakly supervised object detection problem using MIL. The method by Nguyen et al. [19] is a modification of the multi-instance SVM (MI-SVM) formulation of Andrews et al. [2]. It is thus a global-Inter MIL method and has limited ability to cope with intra-class variations because it does not exploit the intra-class distance constraint. In addition, its initial annotation step simply uses entire images as the initially selected instances. This may works for images such as Fig. 1(a), but would fail given more challenging images such as those in PASCAL 2007 dataset [9]. The method proposed by Deselaers et al. [6] employs a conditional random field based formulation to exploit both intra and inter-class distances globally. Unlike our work it ignores the multi-modal problem and is restricted to single viewpoint annotation and detection. Relying on a simplistic detector (SVM) for annotation, their approach uses a fusion of four different feature cues. In addition, quite a few model specific parameters need to be tuned iteratively, including eight parameters that must be learned through a multi-stage grid search on a fully annotated meta training set of other object classes, which can be extremely expensive computationally. In contrast, our method is much easier to implement with only two free parameters. We also do not need a multi-class fully annotated training set for learning. Importantly, our method copes with large intra-class variations explicitly, and critically is able to detect and prevent model drift.

The effectiveness of our approach is validated using the challenging PASCAL 2007 dataset [9]. Our results suggest that both the proposed initial annotation model and model drift detection method contribute to a superior performance in comparison with a number of state-of-the-arts alternative approaches. In particular, we show that, if a strong detection performance can be achieved using a fully-supervised detector then comparable performance can be achieved with the same detector trained using weak supervision.

2. Proposed Method

To learn a detector for a specific object class using WSL, we have a set of positive and negative images of the object without information about its exact location. Taking a MIL approach, each image is considered as a bag, either positive or negative. In our framework (Fig. 2), an initial positive set is first selected using a new initial annotation method. That set is then used to train an object detector in an iterative process. In each iteration, the detector is used as an annotator to select a new positive set (refined object annotation) which is then used to retrain the object detector. The iteration stops when a model drift is detected, at which point the previously learned detector is output as the final detector.

2.1. An Initial Annotation Model

For the initialization step, like previous methods [13, 6], we restrict the possible instances (object locations) per bag (image). For each positive (negative) bag i we select the first n `what-is-object` bounding boxes as instances $x^+_{i,j=1\ldots n}$ ($x^-_{i,j=1\ldots n}$) using the generic object model proposed in [1]. `What-is-object` boxes are only a crude guess on the location of foreground objects and as such a large number of them must be selected to ensure our object of interest is included. We set n to 100 as in [6]. This reduces the number of instances per bag significantly and alleviates the negative influence of background clutter. For each instance an objectness value $o^+_{i,j=1\ldots n}$ ($o^-_{i,j=1\ldots n}$) is obtained as the confidence of the generic object detector [1]. Now the task is to select one instance x^+_{i,j^*} per positive bag i as the initial positive instance. Two metrics are fused to select x^+_{i,j^*}: an intra-class metric $f(x^+_{i,j^*})$ and an inter-class metric $g(x^+_{i,j^*})$. The intra-class metric measures how closely the selected instance x^+_{i,j^*} resembles all other selected instances $x^+_{i=1\ldots N^+,j^*}$. The inter-class metric measures how far a selected positive instances x^+_{i,j^*} is from all negative instances $x^-_{i=1\ldots N^-,j=1\ldots n}$ in the negative bags. The selection of x^+_{i,j^*} is obtained as

$$x^+_{i,j^*} = \arg\max_{x^+_{i,j}} \left[(1 - f_n(x^+_{i,j})) + g_n(x^+_{i,j})\right] \qquad (1)$$

where $j = 1 \ldots n$, and both metrics are normalised:

$$f_n(x^+_{i,j}) = \frac{f(x^+_{i,j}) - \min f(x^+_{i,j=1\ldots n})}{\max f(x^+_{i,j=1\ldots n}) - \min f(x^+_{i,j=1\ldots n})}$$

$$g_n(x^+_{i,j}) = \frac{g(x^+_{i,j}) - \min g(x^+_{i,j=1\ldots n})}{\max g(x^+_{i,j=1\ldots n}) - \min g(x^+_{i,j=1\ldots n})}$$

An initial positive set $\mathcal{C}'$ is then form by the selected x^+_{i,j^*} from each of the n positive bags.

2.1.1 Intra-Class Metric $f(x)$

The instances corresponding to the object of interest should look similar to each other. A candidate set is denoted as $\mathcal{C} = \{c_1, c_2, \ldots, c_{N^+}\}$ composed of one instance per positive bag $\left[c_i \in \left\{x^+_{i,1}, \ldots, x^+_{i,n}\right\}\right]$. The best set $\mathcal{C}^*$ is the one with instances that looks most similar to each other among all possible sets $\mathcal{C}$. This optimal set $\mathcal{C}^*$ is selected by minimizing the intra-class cost function

$$\mathcal{C}^* = \arg\min_{\mathcal{C}} \sum_{c_i \in \mathcal{C}} D\left(c_i, \mathcal{C}_{-i}\right) \qquad (2)$$

where $\mathcal{C}_{-i}$ is set $\mathcal{C}$ excluding c_i. To compute $D\left(c_i, \mathcal{C}_{-i}\right)$, we first sort all instance in $\mathcal{C}_{-i}$ according to their distance to c_i in asending order. Let each instance in this sorted set be c^l_i, we have

$$D\left(c_i, \mathcal{C}_{-i}\right) = (1 - o_i) + \frac{1}{k}\sum_{l=1}^{k} d(c_i, c^l_i) \qquad (3)$$

where k is used to select the k nearest neighbours of c_i, o_i is the objectness value and $d(\ )$ is a distance function to be defined in Sec. 2.1.3. The use of k is similar to the use of k in a k-nearest-neighbour classifier and allows us to handle the case where the set $\mathcal{C}$ is composed of multi-modal data. In particular, when there are multiple modes in the object appearance, caused by e.g. variable viewpoints, the positive instance will naturally form clusters in a feature space. Our formulation, based on measuring distance between k nearest neighbouring instances, naturally encourages instances from multiple clusters being selected. This is in contrast a conventional formulation such as that in [6] where the positive instances are assumed to form a single cluster.

To minimize the cost function (Eq. 2) we employ a genetic algorithm [14], which is an evolutionary algorithm that selects the optimal solution using techniques inspired by evolution. A population of candidate solutions ($\mathcal{C}$s) evolves through reproduction and random mutation towards the optimal solution. In our reproduction step a child $\mathcal{C}^{\text{child}}$ is created from parents $\mathcal{C}^{\text{P1}}$ and $\mathcal{C}^{\text{P2}}$ as follows:

$$\begin{aligned} C^{\text{P1}} &= \{c^{\text{P1}}_1, \ldots, c^{\text{P1}}_m, c^{\text{P1}}_{m+1}, \ldots, c^{\text{P1}}_{N^+}\} \\ C^{\text{P2}} &= \{c^{\text{P2}}_1, \ldots, c^{\text{P2}}_m, c^{\text{P2}}_{m+1}, \ldots, c^{\text{P2}}_{N^+}\} \\ C^{\text{child}} &= \{c^{\text{P1}}_1, \ldots, c^{\text{P1}}_m, c^{\text{P2}}_{m+1}, \ldots, c^{\text{P2}}_{N^+}\} \end{aligned}$$

where m is randomly selected. A mutation probability of P_{mutate} is used to pick bags and the instances from the selected bags are randomly switched.

Given the optimal set $\mathcal{C}^*$, the intra-class metric for each instance $x^+_{i,j}$ is obtained as

$$f(x^+_{i,j}) = D(x^+_{i,j}, \mathcal{C}^*) \quad (4)$$

$$D(x^+_{i,j}, \mathcal{C}^*) = (1 - o^+_{i,j}) + \frac{1}{k}\sum_{l=1}^{k} d(x^+_{i,j}, c^{*l}_i) \quad (5)$$

where $f(x^+_{i,j})$ measures how far an instance $x^+_{i,j}$ is from the ideal set $\mathcal{C}^*$, $o^+_{i,j}$ is the objectness value, c^{*l}_i is the same as c^l_i in Eq. 3 but with respect to $\mathcal{C}^*$.

2.1.2 Inter-Class Metric $g(x)$

An inter-class metric measures how different a positive instance is from all negative instances in the negative bags. To measure the inter-class metric, one needs to compute a representation of the true positive instances, whose locations are unknown, from each positive bag. A straightforward option is to use the entire image [19] to approximate the positive instances, which is apparently inappropriate given objects of large scale change and background clutter. We exploit the nature of `what-is-object` box distribution and a bag-of-words (BoW) representation to come up with a simple positive instance representation, which is the average BoW histograms of all `what-is-object` boxes detected in a positive image. This is based on the observation that `what-is-object` boxes tend to overlap heavily around the correct object of interest and lightly on the background [1]. We illustrate this in Fig. 3 by highlighting the number of time a pixel is included in the top one-hundred `what-is-object` boxes. Fig. 3(a) shows an example where, as a result of this overlapping distribution, features computed on the object of interest are included in most of the `what-is-object` boxes while features computed on the background are only included in few of the `what-is-object` boxes. The average of the BoW histograms over all the `what-is-object` boxes will then be dominated by the features on the object of interest. Note that as shown in Fig. 3(b)&(c), when multiple objects coexisting in an image, an average histogram will be corrupted by other objects as well. Overall, however, enough useful information can be captured as a starting point for representing the positive instances.

We train a SVM to find the boundary between the average histograms (one per positive image) and all the negative instances (n per negative image). To overcome the imbalance in the number of positive and negative instances we employ the negative mining technique of [10]. Normalized histogram intersection is used as the SVM kernel. The inter-class measure $g(x)$ is obtained as the SVM score, which is

(a) (b) (c)

Figure 3. Pixels included in the top 100 `what-is-object` bounding boxes are highlighted. High intensity indicates pixels included in most `what-is-object` bounding boxes.

then fused with the intra-class metric $f(x)$ to select the initial positive set $\mathcal{C}'$ using Eq. 1.

A set $\mathcal{C}^\dagger$ can be obtained as the `what-is-object` window in each positive image with the maximum SVM score. The set $\mathcal{C}^\dagger$ obtained using the global inter-class metric and the optimal set $\mathcal{C}^*$ obtained using the global intra-class metric may also be used as initial annotation of the object class in each training image. We compare sets $\mathcal{C}^\dagger$ and $\mathcal{C}^*$ with the fused result $\mathcal{C}'$ in Section 3.1.

2.1.3 Object Appearance Features

Each instance in a positive or negative image/bag needs to be represented as a feature vector. A standard bag-of-words (BoW) representation is used. We use a regular grid SIFT descriptors [17] and quantize them into 2000 words using k-means clustering. An image window is then represented by a BoW histogram of 2000 dimension.

One limitation of a BoW object appearance representation is that it does not take into account the aspect ratio of the window in which the histogram is computed, which is important to distinguish objects of same class but different viewpoints (Fig. 1). To account for this we define the distance between two image windows (instances) c_i and c_j as

$$d(c_i, c_j) = \frac{\|H_i - H_j\|}{\gamma} + \left(1 - \exp\left(\frac{|R_i - R_j|}{\lambda}\right)\right), \quad (6)$$

where H_i is the BoW histogram of c_i, R_i is its window aspect ratio and γ and λ are normalization factors. We obtain γ for each class as the median distance $\| H_i - H_j \|$ of all pairwise combination of instances in the positive bags in a training set. λ is obtained by fitting an exponential over the histogram of $| R_i - R_j |$ on all pairs of instances in positive bags in all classes.

2.2. Detector Learning

After an initial positive set is obtained, a detector is trained iteratively. We use the part-based detector of [10] because it is designed to address the multi-modal problem in object detection by learning a mixture of part-based models for different modes. After training the part-based detector using the initial positive instances $\mathcal{C}'$, the detector is then run on each positive training image and the image window

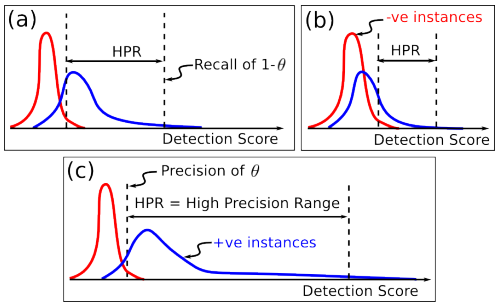


Figure 4. Illustration of model drift detection. See text for explanation.

with the highest detection score is selected as the new positive window. The detector is then retrained with the new set of positive windows. At each iteration, model drift is measured to determine whether the iteration should be terminated. If no model drift is detected, the iteration will terminate when the annotation result converges, i.e. same set of instances are selected.

2.3. Model Drift Detection

Similar to adaptive object tracking, model drift can amplify itself resulting in model failure. Detecting model drifting is challenging in WSL of object detectors. This is because without knowing where the positive object instances are located in each positive bag, it is impossible to measure directly whether a stronger object detector has been obtained after each iteration. It is possible to measure that indirectly by utilising only the weak labels for the training data. This is achieved through measuring the change in the difference between the detector score distribution on the positive instances selected by the current model, and that on negative instances in negative bags over consecutive iterations.

More specifically, in the first iteration the initial annotation (Sec. 2.1) used for the positive set would not be very accurate because no detector is involved in the initial annotation model. We thus expect the detector score distribution for the positive set $\mathcal{C}'$ to have a fair amount of overlap with that of the negative set composed of random sampled instances from the negative bags. Note that the latter is accurate whilst the former is only an approximation as $\mathcal{C}'$ inevitably contains inaccurate annotations. The two distributions can be estimated by constructing histograms of the detector scores. Typical distributions of the two sets are illustrated in Fig. 4(a).

After each iteration if the model is getting stronger, we expect the detector to favour those instances that appear similar to the high scoring positive instances from the previous iteration. This will affect the positive and negative detection scores in that there will be less overlap between the two distributions and a higher separation between the positive and negative detector score as illustrated in Fig. 4(c). In contrast, when the model starts to drift towards some other objects or common background areas in the positive images, the high scoring positive instances from the previous iteration will no longer have high detector scores. This will result in higher overlap between the two distributions as illustrated in Fig. 4(b). We can thus detect model drift by monitoring how the overlap between the two distributions changes over iterations.

More precisely, we measure the overlap using a high precision range (HPR) measure, as illustrated in Fig. 4. HPR is computed as:

$$HPR = S^{+}_{1-\theta} - S^{-}_{\theta} \tag{7}$$

where $S^{+}_{1-\theta}$ is the detection score of the selected positive instances at a recall rate of $1-\theta$ and S^{-}_{θ} is the score of negative instances at a precision value of θ. θ is a value that should be set close to 1. Obviously both the recall and precision rates are computed by assuming that the positive instances selected by the current model are correct. Since this assumption is generally invalid in practice, both $S^{+}_{1-\theta}$ and S^{-}_{θ} are only approximations. However, since they are computed from distributions estimated from large number of instances, they are fairly robust against annotation errors as suggested by our experiments.

3. Experiments

We evaluate our approach on its ability both to automatically annotate a training set and to detect objects of interest in a testing set.

Datasets and settings – We used two versions of the PASCAL VOC2007 dataset [9]: *AllView* and *SingleView*. *AllView* is used to test our framework's ability to handle multi-modal (multi-viewpoints) data and contains all 20 classes of the PASCAL dataset *without* manual viewpoint annotation. As far as we know this is the first time results on data of mixed viewpoints are reported. *SingleView* is used to compare our approach with the approach of [6] which is difficult to implement and only reports results on single viewpoint data. *SingleView* dataset uses the *left* and *right* views of the classes aeroplane, bicycle, boat, bus, horse, and motorbike, for a total of 12 classes. For all our experiments, we fixed our nearest neighbour parameter in Eqs. 3 and 5 to $k = 5$ and θ in Eq. 7 to 0.95. We found the results are insensitive to both parameters. The detector is trained using the default parameters of [10]. For the *SingleView* dataset we trained a single component model and for the *AllView* dataset we trained a three component model. For annotation we report the correct detection rate in the training set and for detection we report the average precision (AP) on the testing set as defined by the PASCAL challenge [9]. For both annotation and detection an object is considered as correctly local-

Figure 5. Example of training image annotation results.

ized/detected if the overlap is greater than 50%.

Competitors – We implemented the approach of Nguyen et al. [19] for comparison on the *AllView* datasets. Since [19] is based on MI-SVM approach of Andrew et al. [2], we also implemented a MI-SVM approach modified for object detection using `what-is-object` windows for defining instances in each image. For annotation on the *SingleView* set we compare with the results reported in [6]. [6] provides average detection rate for their approach as well as the approaches of [20] and [3].

3.1. Automated Annotation Results

Comparing with [19] and [2] on *AllView* – Table 1 shows that averaged across all 20 classes our method achieves a performance increase of 36% compared to the method in [19]. The main reasons for the poor performance of [19] are due to 1) using entire image to initialize the iterative training, 2) ignoring intra-class constraint thus not be able to cope with multi-modal distribution, and 3) no model drift detection. An average histogram of `what-is-object` windows is used in the initialization step of both MI-SVM [2] and our inter-class metric, however, our performance is still 20% better than [2] because of the contribution of our global intra-class metric and model drift detection. Examples of annotation results using the different methods are given in Fig. 5. In particular, the first two images of Fig. 5 shows that the proposed method is able to detect objects at different viewpoints more accurately than the two compared methods. As seen in Fig. 5 bottom left, if the object of interest occupies the entire image, [19] has the best performance because it is biased toward choosing the full image as the object of interest. Fig. 5 bottom middle, shows that when the object of interest is relatively large and is the only foreground object, MI-SVM [2] has the best performance as most of the `what-is-object` boxes will be distributed greatly over the single foreground object. Finally if the object of interest is very small and low contrast, like the sheep in Fig. 5 bottom right, all three methods fail.

Effectiveness of model drift detection – Table 1 shows the effectiveness of model drift detection in our method; with

Class	Our Initialization			Our Final Annotation		Competitors	
	Intra $\mathcal{C}^*$	Inter $\mathcal{C}^\dagger$	Intra + Inter $\mathcal{C}'$	Detector	Detector + Model Drift	MI-SVM [2]	Nguyen et al. [19]
aeroplane	31.1	41.2	45.4	42.4	42.4	37.8	30.7
bicycle	18.5	17.7	20.6	49.0	46.5	17.7	16.5
bird	25.2	28.2	29.7	18.2	18.2	26.7	23.0
boat	13.8	13.3	12.2	08.8	08.8	13.8	14.9
bottle	03.3	05.3	04.1	02.5	02.9	04.9	04.9
bus	31.2	35.5	37.1	40.9	40.9	34.4	29.6
car	26.7	33.0	41.0	49.0	73.2	33.7	26.5
cat	42.1	50.1	53.4	35.0	44.8	46.6	35.3
chair	06.7	05.4	06.5	07.4	05.4	05.4	07.2
cow	28.4	30.5	31.9	28.4	30.5	29.8	23.4
diningtable	24.0	16.0	20.5	16.5	19.0	14.5	20.5
dog	33.3	36.1	40.9	27.1	34.0	32.8	32.1
horse	30.7	38.7	37.3	48.8	48.8	34.8	24.4
motorbike	34.7	44.1	46.5	65.3	65.3	41.6	33.1
person	14.3	20.6	22.3	08.2	08.2	19.9	17.2
pottedplant	09.4	11.4	10.2	10.6	09.4	11.4	12.2
sheep	26.0	25.0	27.1	16.7	16.7	25.0	20.8
sofa	25.3	23.6	32.3	32.3	32.3	23.6	28.8
train	42.9	47.9	49.0	54.8	54.8	45.2	40.6
tvmonitor	10.6	08.6	09.8	05.5	05.5	08.6	07.0
Average	**23.9**	**26.6**	**28.9**	**28.4**	**30.4**	**25.4**	**22.4**

Table 1. Detailed training set annotation results (%) for *AllView*.

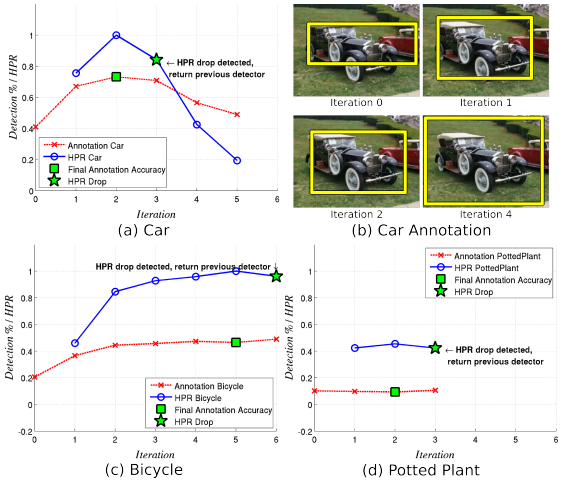


Figure 6. Model drift detection. (a),(c),(d) Plots of annotation and HPR vs iteration for the car, bicycle and potted plant classes. (b) Annotation box change over different iterations for car.

model drift detection a noticeable increase in the annotation accuracy is observed. When comparing our approach with and without model drift detection, we see model drift occurred in 9 out of the 20 classes and our model drift detection has a positive impact on 6 of them. Importantly, the influence of model drift detection is drastic in classes such as car where the accuracy increases from 49% to 73%.

To gain some insight into our model drift detection we plot the annotation accuracy and HPR values against the iteration number in Fig. 6. The car class (Fig. 6(a)) clearly demonstrates the need for model drift detection. The annotation accuracy of the car class increases in the first two iterations then starts to decrease. Visually the detector

	Intra + Inter $\mathcal{C}'$	Detector + Model Drift	Deselaers et al. [6]	*Russel et al. [20]	*Chum et al. [3]
Average	**40**	**49**	**50**	**20**	**29**

** As reported in [6]*

Table 2. Average training set annotation results (%) for *SingleView*.

bounding boxes starts to converge to the correct localization then diverge to include the background scene as shown in Fig. 6(b). In this case we want to stop the iterations when the model starts drifting from the correct localization. We can see from Fig. 6(a) our HPR measure has a linear relationship with the annotation accuracy, as a result when the HPR measure starts to decrease our approach correctly predicts that the model is starting to drift and stops the iterative learning. In some cases such as of bicycle and pottedplant (Fig. 6(c) and 6(d)), both the annotation accuracy and the HPR measure begin to converge to a constant value. When this occurs the sign of the change in HPR over consecutive iteration is not reliable because in effect we are doing a derivative of a noisy line with zero slope. As a result, our model drift prediction is not guaranteed to stop when the annotation accuracy is at a high value. However, we can see that in these cases the difference in annotation accuracy when the iteration begin to converge is small compare to the increase seen at the start. Therefore, the error caused by our model drift prediction failure will be small.

Effectiveness of fusing intra and inter-class metrics – Table 1 also shows the effectiveness of fusing the intra and inter-class metrics for initial annotation. A noticeable improvement is obtained when the two metrics are combined. To give some insight, we illustrate some of the initial annotation results in Fig. 7. Inter and intra metrics provide different annotation bounding boxes as seen in Fig. 7. Fig. 7 Column 1 shows an example where by fusing the two metrics, correct annotation is obtained even when each metric alone selects a wrong instance. When one of the inter or intra-class metric annotates the object correctly the fusion is weighted towards the correct metric (Fig. 7 Column 2). However, the downside of fusing the metrics, as illustrated in Fig. 7 Column 3, happens when one of the metrics selects the wrong instance with a high confidence value.

Comparing with [6] on *SingleView* – We compare our approach with that of Deselaers et al. [6] which only reports results on the *SingleView* dataset. Table 2 summarizes the annotation results of our algorithm as well as two other methods [20] and [3] reported in [6]. It shows that model drift detection improve the annotation accuracy. Our annotation accuracy is nearly double that of Russel et al. [20] and Chum et al. [3] and is within 1% of Deselaers' approach [6]. Note that this is achieved without fusing more than one feature, and tuning fewer parameters. More importantly we do not need a meta training data set consisting of multiple classes of fully annotated images for parameter tuning.

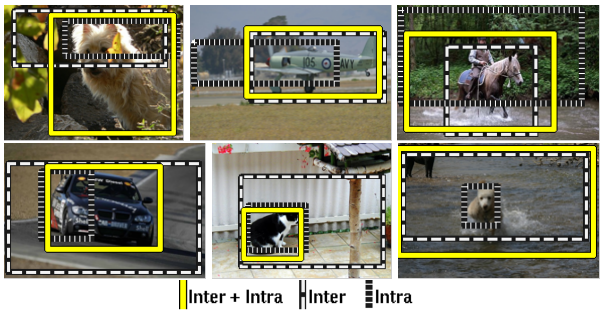


Figure 7. Effectiveness of fusing intra-class and inter-class metrics for initial annotation.

3.2. Detection Results

High annotation accuracy on training images will not matter if it does not result in good detection performance. After all it is the detection performance on unseen images that one is after. Table 3 compares the detection average precision (AP) of our detector trained with weakly supervised (WS) training set vs the detector trained with fully supervised (FS) training set on the VOC2007 test data. Our WSL approach is able to achieve a mean AP, over the 20 classes, of 0.139 compared to a mean AP of 0.263 for the FS approach; that is, our WS approach's performance is about 53% of the FS approach. More interestingly we see that 4 of the classes, train, motorbike, car, and bird, achieve an AP that is nearly the same as the FS detector, that is, over 95% of the FS detector performance with an average annotation accuracy of 53%. For the motorbike class, our WS detector even beats the FS detector with a 65.3% annotation accuracy. This suggests that manual annotation does not necessarily creates the best data for detector learning. Furthermore, a total of 6 classes, achieved a performance greater than 80% of the FS detector with an average annotation accuracy of 50%. In general, it is noted that when the FS detector performs strongly, the gap between a WS detector and a FS detector becomes smaller (see Fig. 8). Similar phenomenon is also observed by Deselaers et al. [6], but only restricted to single-viewpoint cases. Specifically their WS detector achieved an AP of 48% of that of FS detector on 6 selected classes for which the FS detector performed well. The tvmonitor class is an exception, the FS detector performs very well on this class but the WS detector performs poorly. The main reason for this is that our initial annotation algorithm performs very poorly on the tvmonitor class (Table 1) because SIFT descriptors can not effectively model the smooth rectangular structure of tvmontiors. Some examples of detection results are shown in Fig. 9.

This is a very encouraging result. This shows that when a FS detector performs reasonably well for a class, i.e. a AP of around 0.45, there is little gap between the FSL approach

	aeroplane	bicycle	bird	boat	bottle	bus	car	cat	chair	cow	diningtable	dog	horse	motorbike	person	pottedplant	sheep	sofa	train	tvmonitor	**Average**
WSL	.134	.440	.031	.031	.000	.312	.439	.071	.001	.093	.099	.015	.294	.383	.046	.001	.004	.038	.342	.000	**.139**
FSL [10]	.290	.546	.006	.134	.262	.394	.464	.161	.163	.165	.245	.050	.436	.378	.350	.088	.173	.216	.340	.390	**.263**

Table 3. Weakly supervised vs fully supervised learning detection results for *AllView*.

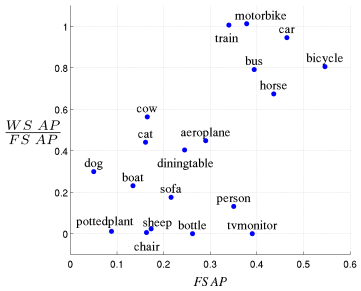


Figure 8. Weakly supervised average precision / Fully supervised average precision vs fully supervised average precision.

Figure 9. Detection results on PASCAL 2007 test data.

and our WSL approach. Since strong detectors are being improved constantly, one would expect the effectiveness of our approach to become more obvious as stronger detectors become available. In addition, there may exist different detectors that perform better on different object classes. Our approach can also incorporate different detectors for different classes to take advantage of their different strengths.

4. Conclusion

We presented a framework for the challenging task of training an object detector from weakly labelled data. Our framework includes a novel initialization routine that is able to provide a good initial annotation of the training set to start iteratively training an object detector. The iterative learning of the detector is stopped automatically by a new mechanism for determining when the detector is about to drift away from the objects of interest. For some of the PASCAL 2007 classes, our weakly supervised learning framework is able to train a detector that can achieve a detection accuracy comparable to that of a fully supervised detector.

References

[1] B. Alexe, T. Deselaers, and V. Ferrari. What is an object? In *CVPR*, pages 73 – 80, 2010. 3, 4

[2] S. Andrews, I. Tsochantaridis, and T. Hofmann. Support vector machines for multiple-instance learning. In *NIPS*, pages 577–584. 2003. 2, 6

[3] O. Chum and A. Zisserman. An exemplar model for learning object classes. In *CVPR*, pages 1 – 8, 2007. 1, 6, 7

[4] S. Clark, J. R. Curran, and M. Osborne. Bootstrapping pos taggers using unlabelled data. *CoNLL-03*, 2003. 1

[5] D. J. Crandal and D. P. Huttenlocher. Weakly supervised learning of part-based spatial models for visual object recognition. In *ECCV*, 2006. 1, 2

[6] T. Deselaers, B. Alexe, and V. Ferrari. Localizing objects while learning their appearance. In *ECCV*, volume 6314, pages 452–466, Jan 2010. 1, 2, 3, 5, 6, 7

[7] T. Deselaers and V. Ferrari. A conditional random field for multiple-instance learning. In *ICML*, 2010. 2

[8] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Perez. Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence*, 1997. 1

[9] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. 2, 3, 5

[10] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *TPAMI*, 32(9):1627 – 45, 2010. 1, 4, 5, 8

[11] R. Fergus, P. Perona, and A. Zisserman. Weakly supervised scale-invariant learning of models for visual recognition. *IJCV*, 2007. 1, 2

[12] Z. Fu, A. Robles-Kelly, and J. Zhou. MILIS: Multiple Instance Learning with Instance Selection. *TPAMI*, (99), 2010. 1, 2

[13] C. Galleguillos, B. Babenko, A. Rabinovich, and S. Belongie. Weakly supervised object localization with stable segmentations. In *ECCV*, Jan 2008. 1, 2, 3

[14] D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st edition, 1989. 3

[15] H. Grabner, C. Leistner, and H. Bischof. Semi-supervised on-line boosting for robust tracking. In *ECCV*, 2008. 1

[16] C. Leistner, A. Saffari, and H. Bischof. Miforests: multiple-instance learning with randomized trees. In *ECCV*, 2010. 2

[17] D. G. Lowe. Object recognition from local scale invariant features. In *ICCV*, volume 2, pages 1150 – 1157, 1999. 4

[18] O. Maron and T. Lozano-Perez. A framework for multiple-instance learning. In *NIPS*. 1998. 2

[19] M. H. Nguyen, L. Torresani, F. de la Torre, and C. Rother. Weakly supervised discriminative localization and classification: a joint learning process. In *ICCV*, pages 1925 – 1932, 2009. 1, 2, 4, 6

[20] B. Russell, W. Freeman, A. Efros, J. Sivic, and A. Zisserman. Using multiple segmentations to discover objects and their extent in image collections. In *CVPR*, volume 2, pages 1605 – 1614, 2006. 1, 6, 7