Electronic Letters on Computer Vision and Image Analysis 15(1):77-99, 2016

Efficient Labelling of Pedestrian Supervisions

Kyaw Kyaw Htike

School of Information Technology, UCSI University, Kuala Lumpur, Malaysia

Received 7th Mar 2016; accepted 26th Jun 2016

Abstract

Object detection is a fundamental goal to achieve intelligent visual perception by computers due to the fact that objects are the basic building blocks to achieve higher level image understanding. Among the numerous categories of objects in the real-world, pedestrians are among the most important due to several potential benefits brought about by successful pedestrian detection. Often, pedestrian detectors are trained in state-of-the-art systems using supervised machine learning algorithms which necessitates costly and often tedious manual annotation of pedestrians in the form of precise bounding boxes. In this paper, a novel weakly supervised learning algorithm is proposed to train a pedestrian detector that requires, instead of bounding boxes, only annotations of estimated centres of pedestrians. The algorithm makes use of a *pedestrian prior* learnt in an unsupervised way from the input video and this prior is fused with the given weak supervision information in a systematic manner. By evaluating on publicly available datasets, we demonstrate that our weakly supervised algorithm reduces the cost of manual annotation of pedestrians by more than four times while achieving similar performance to a pedestrian detector trained with standard bounding box annotations.

Key Words: Pedestrian detection, weak supervision, unsupervised prior, cue fusion.

1 Introduction

Object detection is a critical component in an attempt to make computers automatically understand and interpret visual data. Among the numerous classes of objects that exist in real-life scenes, pedestrians are among the most crucial, due to the fact that being able to automatically detect and reason about pedestrians has enormous benefits and far-reaching applications in the fields of computer vision, machine learning and Artificial Intelligence in general.

To detect pedestrians in an image, a trained classifier is used to score each image patch corresponding to the multi-scale sliding windows and the local modes of the score space indicate the locations and the spatial extent of pedestrians in the image [1, 2, 3, 4, 5]. Most pedestrian detectors are trained in a supervised way in which the training dataset (which consists of a sufficient number of pedestrian and non-pedestrian examples) is provided by the user. The training is achieved by framing the problem as *learning* a model for a binary classification task (*i.e*. to differentiate between the class of pedestrians and the class of non-pedestrians).

Correspondence to: ali.kyaw@gmail.com

Recommended for acceptance by Frédéric Lerasle
http://dx.doi.org/10.5565/rev/elcvia.881
ELCVIA ISSN: 1577-5097
Published by Computer Vision Center / Universitat Autònoma de Barcelona, Barcelona, Spain

Constructing the training dataset requires manual annotation of pedestrians and non-pedestrians. In state-of-the-art research, the manual annotation of pedestrians is often given in the form of bounding boxes tightly fitting the pedestrians [6, 7, 8, 9, 10, 11, 12, 13]. To extract non-pedestrian training examples, images that do not contain any pedestrians are typically supplied.

This pedestrian annotation work is highly expensive in terms of cost, labour and time due to the following reasons:

- Accurately annotating each pedestrian requires thought and focus because it requires strictly adhering to a set of guidelines (such as on what defines the extent of a pedestrian, what parts of the pedestrian the bounding box should cover, and how much of the surrounding background the bounding box should include as context).
- The attention and effort for the above have to be repeated and sustained for thousands or hundreds of thousands of pedestrians. Nowadays, certain datasets in various fields of computer vision can even go up to millions of data points.
- The annotation task is inherently uninteresting and stressful due to the fact that there is little margin for error or deviation from the expected guidelines and even the provided guidelines could often be ambiguous. Any errors introduced is likely to negatively affect the quality of the training dataset (and which will in turn affect the resulting trained classifier).

Rather than using bounding box annotation required in most state-of-the-art research [6, 7, 8, 9, 10, 11, 12, 13], we propose, in this paper, a novel weakly supervised learning algorithm for training pedestrian detectors that requires only *estimates* of centres of pedestrians (as shown in Figure 1).

This allows for an easier and faster annotation compared to bounding box annotation. In particular, this type of annotation has the following benefits:

- Performing each annotation is very fast; in fact, it just takes one click of the mouse.
- Even a single click does not need to be accurate. This requires much less attention and effort.
- Owing to the lower stress and higher fun factor, the annotation work can be made more pleasant for people, *e.g*. it can be given to people as a game: the one who can tick the most pedestrian centres (roughly) in a given time wins the game. This can be used to help acquire large amounts of data for training with much less cost.

Our algorithm makes use of a *pedestrian prior* learnt in an unsupervised way from video and this prior is *fused* with the given weak supervision information in a principled manner.

We show on publicly available datasets that, despite the weak supervision, our algorithm performs comparably with bounding box supervision (termed in this paper as *strong supervision*) despite having a much lower cost (measured in terms of the time it takes to complete the annotation). To be more precise, our weakly supervised algorithm reduces the cost of manual annotation by over four times while achieving similar performance as pedestrian detectors trained with bounding box annotations.

2 Background

In this section, we give an overview and compare and contrast two related areas, namely weakly supervised learning and semi-supervised learning. This is necessary because of two main reasons:

- Firstly, these two areas can cause confusion among researchers and the difference between the two areas is sometimes not appreciated.

Figure 1: Strong versus weak annotation (best viewed in colour). On the left is the standard way of annotating pedestrians for training a pedestrian detector. On the right is the weak supervision (only approximate centres of pedestrians) required by our proposed algorithm. Note that pedestrians are of different sizes in the video due to projective distortion and hence our algorithm has to cope with *both* noisy locations and unknown scales. Weak supervision on the right is much faster and easier for a human annotator than the strong supervision on the left. Images shown are from the CUHK Square dataset [14].

- Secondly, although there have been quite a lot of work that has been done semi-supervised learning, weakly supervised learning is relatively recent (hence the need for this background section to clarify the terminology).

Firstly, we start by formalising the (standard) supervised learning for binary classification. Given the training data

$$\{\mathbf{x}_1, \ldots, \mathbf{x}_N\},$$

where $\mathbf{x}_i \in \mathbb{R}^k$ is a feature vector and the corresponding labels

$$\{y_1, \ldots, y_N\},$$

where $y_i \in \{1, 0\}$ is the supervision label associated with $\mathbf{x}_i$, training a classification model correspond to optimizing the following objective function:

$$\mathcal{M}_{\text{trained}} = \arg\min_{\mathcal{M}} \sum_{i=1}^{N} f_L(\mathcal{M}, \mathbf{x}_i, y_i) + \alpha f_R(\mathcal{M}), \tag{1}$$

where $\mathcal{M}_{\text{trained}} \in \mathbb{R}^k$ is the trained model, f_L is the *loss function*, f_R is the *regularization* function to penalize $\mathcal{M}$ of higher complexity and α is the trade-off term between the regularization term and loss term.

The $\mathcal{M}$ is general and can be any classification model. For instance, if the model is a linear (regularized) classifier, Equation 2 can be written as:

$$\mathbf{m}_{\text{trained}} = \arg\min_{\mathbf{m}} \sum_{i=1}^{N} f_L(\mathbf{m}, \mathbf{x}_i, y_i) + \alpha f_R(\mathbf{m}), \tag{2}$$

where $\mathbf{m}_{\text{trained}} \in \mathbb{R}^k$ is the vector of linear weights for the trained classifier. Furthermore, f_L, f_R and α are the same as discussed previously. For L2-regularization, we have $f_R = \mathbf{m}^T\mathbf{m}$ which penalises large values of $\mathbf{m}$ and for L1-regularization, $f_R = [1, \ldots, 1]^T\mathbf{m}$ which encourages sparse solutions. For SVM, the loss function is:

$$f_L(\mathbf{m}, \mathbf{x}_i, y_i) = \max(0, 1 - y_i \mathbf{m}^T \mathbf{x}_i) \tag{3}$$

and for logistic regression,

$$f_L(\mathbf{m}, \mathbf{x}_i, y_i) = \log(1 + \exp(-y_i \mathbf{m}^T \mathbf{x}_i)). \tag{4}$$

At test time, for SVM, the classifier score s_t of a test feature vector $\mathbf{x}$ is given by

$$s_t = (\mathbf{m}_{\text{trained}})^T \mathbf{x}, \tag{5}$$

whereas for logistic regression, the score is given by

$$s_t = \frac{1}{1 + \exp(-(\mathbf{m}_{\text{trained}})^T \mathbf{x})}. \tag{6}$$

Semi-supervised learning, in many aspects, is similar to standard supervised machine learning except that in semi-supervised learning, the majority of training data are *unlabelled*. But both the labelled data and unlabelled data are still assumed to come from the same distribution and for each labelled data, the supervision is *not* weak, *i.e*. the supervision (strength) is the same as the standard supervision.

Formally, in semi-supervised learning, the dataset can be written as $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ and $\mathbf{Y} = \{y_1, \ldots, y_N\}$ and most of $y_i \in \mathbf{Y}$ are unknown and can be considered as *latent* variables. The model (parameters), in addition to these latent variables, are learnt during the training stage. Thus, due to having more unknown variables to solve and also because of the fact that the optimization is prone to get stuck in poor local optima, semi-supervised learning is a harder problem to solve than supervised learning.

Weakly supervised learning, on the other hand, is a machine learning paradigm distinct from supervised or semi-supervised learning. In weakly supervised learning, as the name suggests, the training data are only *weakly* annotated. For example, rather than being given exact patches of objects, the algorithm may be given only weak indications of the presence of objects in images.

However, it is to be noted that in weakly-supervised learning, even though each supervision is weak, *all* the training data are *still* given (these) weak supervisions, unlike in semi-supervised learning where there are examples that are not supervised (*i.e*. the *unlabelled* examples).

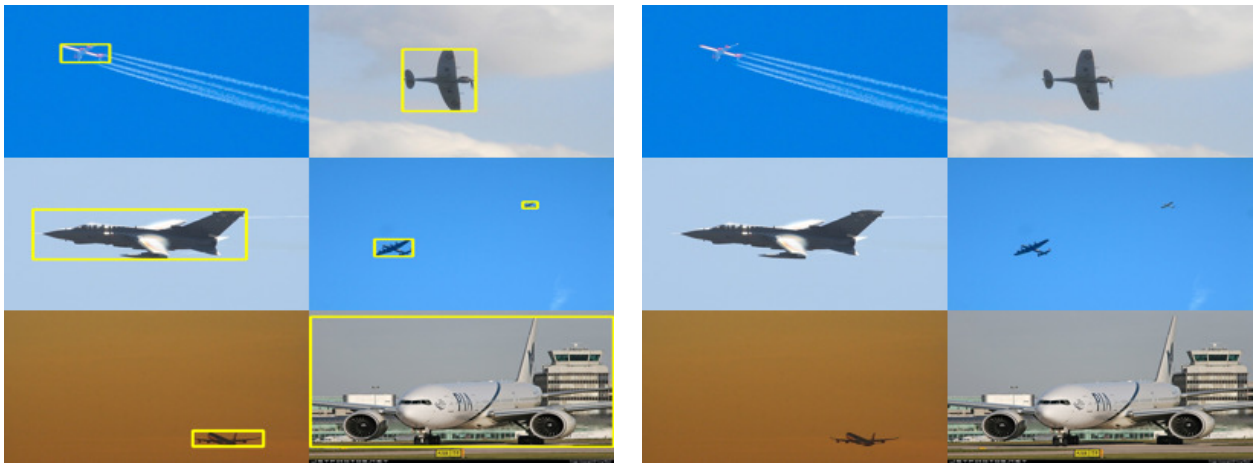
The weakly supervised learning problem (for binary classification) can be formalised as follows. Let the training dataset be made up of k groups (or *bags*):

$$\{(\mathbf{X}_1, y_1), (\mathbf{X}_2, y_2), \ldots, (\mathbf{X}_k, y_k)\},$$

where $y_i \in \{1, 0\}$. A bag $\mathbf{X}_i$ consists of several data *instances*, *i.e*. $\mathbf{X}_i = \{\mathbf{x}_{i1}, \mathbf{x}_{i2}, \ldots\}$ where $\mathbf{x}_{ij}$ is a feature vector belonging to either the positive or negative class (for binary classification). However, the labels of the instances are *not* observed (*i.e*. they are latent variables). Instead, the entire bag is associated with a *single* bag label which can be either a *positive bag* if $y_i = 1$ or a *negative bag* if $y_i = 0$. Each positive bag contains *at least* one positive instance and each negative bag contains *only* negative instances.

In order to make it more general, although labels are not given for each instance, in each positive bag, the probability distribution of the instances belonging to the positive class may be given, *i.e*. in a positive bag, some instances may be more likely to be from the positive class than others. If no such (prior) information is available, then a uniform distribution is assumed, *i.e*. in a positive bag, all the instances are equally likely to be from the positive class.

As an example of weakly supervised training, consider training an aeroplane detector. Figure 2 illustrates a comparison between annotation requirements for standard supervision and weak supervision in the context of training the aeroplane detector. Instead of being given patches corresponding to aeroplanes (*i.e*. bounding boxes of aeroplanes in the training dataset), we may only be given images containing aeroplanes without any information about the exact location and extent of aeroplanes in each image; the only cue that is given is that



(a) Standard supervision (b) An example of weak supervision

Figure 2: An example of the comparison between standard supervision and weak supervision. Weak supervision requires less human effort than standard ("strong") supervision. Note that negative images (*i.e.* images that do not contain any aeroplanes) are not shown because they are the same for both supervised and weakly-supervised learning cases and they are relatively cheap to obtain.

each training image contains one or more aeroplanes. This is a weak label as opposed to a strong label in the form of exact bounding boxes of aeroplanes.

In the context of the formalisation described previously, in this case, each training image (regardless of whether the image contains an aeroplane or not) can be considered as a bag and each image shown in Figure 2(a) can be considered a positive bag. Each positive bag would then consist of a set of feature vectors corresponding to all the possible patches (*i.e.* instances) at various locations and scales of the corresponding image. However, in a positive bag, only one of those patches would truly be the patch corresponding to an aeroplane but this information is not known. As can be observed, without any prior information about which patches are more likely to be positive instances in each bag, it is very computationally expensive to jointly infer where an aeroplane is in each positive training image from the space of all possible patches in all the training images.

As noted earlier, weakly-supervised learning is different from semi-supervised learning in that in semi-supervised learning, although we have a small number of labelled data (and a lot of unlabelled data), for the labelled data that is available, the supervision is not weak.

3 Related work

Compared to training pedestrian detectors using strong (*i.e.* stardard) supervision, the literature concerning weakly supervised training is fairly limited. Furthermore, most of the literature on weakly supervised learning in images use a different setting than our proposed approach.

In the existing approaches, supervision is given in the form of image-level labels where the exact locations and spatial extents 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 is by formulating it as a Multiple Instance Learning (MIL) problem [15, 16] 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 contain the desired object [17, 18]. The performance of such a system, however, depends heavily on the results of segmentation.

Furthermore, in most existing approaches, datasets are assumed in which an object occupies a large central portion of each image in most of the training images [19, 20, 18, 17]. Although attempts have been made (*e.g*. [21]) to make datasets more challenging, the observation made above still largely applies. This is in contrast to our approach 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 very small portion of the frame.

Deselaers *et al*. [22] propose an iterative algorithm to learn object classes from weakly labelled images using a conditional random field that progressively adapts to the new classes. Chum and Zisserman [19] 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 a SVM classification objective function, Nguyen *et al*. [20] optimizes the function to infer the variables. Weakly supervised learning is tackled as a structured output learning framework in [23]. All of the aforementioned approaches deal only with images and do not make use of information that can be exploited in surveillance-type videos.

Prest *et al*. [24] propose a weakly supervised learning approach for YouTube video clips. Their approach, which is essentially an extension of [22] to video, is fundamentally different from this paper 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, whereas we do not have any such assumption and setting (which can be quite restricting) and we are dealing with long videos captured by a static uncalibrated camera overlooking a scene.

Bilen *et al*. [25] incorporate domain-specific prior knowledge (such as mutual exclusion and symmetry) in a posterior regularization setting with a softmax margin learning framework for object detection. Even though this approach can handle a limited number of multiple objects in each image, it requires careful formulation of the prior knowledge and the optimization is challenging and can be sensitive to local optima.

In order to minimise the problem of local optima solutions, Bilen *et al*. [26] propose a system that learns a set of exemplars using convex clustering and then favouring solutions that are similar to these exemplars. This implicitly enforces a prior that encodes the high similarity (*i.e*. low intra-class variability) between objects of the same class.

Instead of weak supervision at the level of objects, we may also have weak supervision at the level of object parts. An example of this is the Deformable Part Models by Felzenszwalb *et al*. [10, 8]. However, this line of work is different from ours in that the goal of our work is about weak supervision at the (holistic) object level (*not* given any bounding boxes for the object), not at the object part level (given bounding boxes for the object).

Recently, convolutional neural networks (CNNs) [27, 28] have been used to help with weakly supervised learning. Oquab *et al*. [29] show that a CNN that has been trained for the purpose of object classification can not only perform the intended classification task but also can be utilised for predicting estimates of object locations. A major limitation of the algorithm is that it outputs only rough locations and not extents of objects. Moreover, this method shares limitations with many other works (such as [19, 20, 18, 17]) in that they assume that the object in interest in each image occupies the central portion of the image.

4 Contributions

The key contributions for this paper are as follows:

1. A weakly supervised training algorithm that makes use of approximate centre location annotation for training pedestrian detectors for videos.

2. Unsupervised learning of a pedestrian prior for a given video.

3. Combining cues from the unsupervised learnt prior and weak supervision in an optimization framework.

4. The algorithm works with low resolution videos that do not allow accurate part-based modelling and discovery, and that have multiple objects of varying sizes in each frame.

5. The algorithm is not sensitive to low-level segmentation unlike many state-of-the-art weak-supervision approaches using Multiple Instance Learning.

6. The approach is efficient since it does not require jointly solving all the weak supervisions.

5 Our Approach

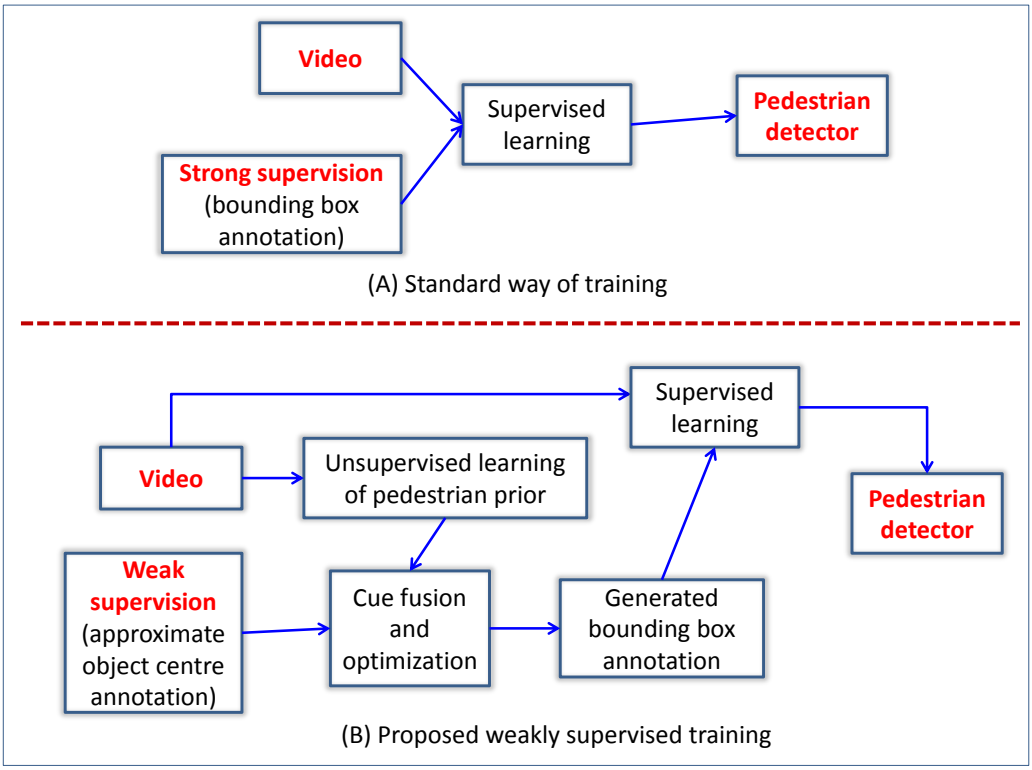


Figure 3: (A) shows the standard way of training pedestrian detectors. In comparison, (B) illustrates the overview of our proposed algorithm.

An overview of the algorithm is illustrated in Figure 3. Let $\mathbb{V} = [I_1, I_2, \ldots, I_N]$ be a given video of N frames. Let $\mathbb{S}_{\text{weak}} = \{\mathbf{cc}_1, \mathbf{cc}_2, \ldots, \mathbf{cc}_M\}$ be a set of M given weak supervisions, where a weak supervision, $\mathbf{cc}_i = [cc_i^x, cc_i^y, cc_i^n]$, is a three element vector where cc_i^n gives the frame number in $\mathbb{V}$ associated with $\mathbf{cc}_i$, and cc_i^x and cc_i^y give the x-coordinate and y-coordinate (in image-plane) respectively corresponding to the approximate centre location of a pedestrian in frame cc_i^n of $\mathbb{V}$.

The goal is to obtain a pedestrian detector given only $\mathbb{S}_{\text{weak}}$ without being provided any bounding box annotations (which the traditional supervised training requires). Our algorithm is made up of three distinct stages.

In the first stage, we learn a *pedestrian prior* in an unsupervised way using knowledge that can be automatically extracted from $\mathbb{V}$.

To be more specific, for any video captured using a static uncalibrated camera, the dynamic background of the scene can be effectively modelled and foreground objects can be detected and extracted.

Although these extracted foregrounds are usually very noisy, by letting the system observe a sufficiently long duration of $\mathbb{V}$, considering all the extracted foregrounds jointly and *generalising* over them (in an offline

process), the system is able to get a general idea and automatically learn about interesting objects in the scene associated with $\mathbb{V}$ without even attempting to explicitly distinguish between different object categories.

This gives us a scene-specific *general object classifier* which could also be from another perspective be interpreted as a *pedestrian prior*.

More specifically, the pedestrian prior can be represented as $P(\text{pedestrian}|\text{patch})$, *i.e.* given any patch in $\mathbb{V}$, the pedestrian prior gives the *prior probability* that the given patch depicts a pedestrian.

We digress here to clarify about the term "prior probability". In the context of Bayesian statistics, prior probability distribution gives the probability distribution before observing any data (*i.e. evidence*). Observing the data provides the evidence (which in turn provides the *likelihood*) and the prior distribution is then weighted by the evidence.

To be more accurate, the regions of the prior probability distribution that are supported by the evidence are given higher weights. The outcome of this is the *posterior probability* distribution which is the combination of the prior distribution (or *belief*) and the evidence. Informally, it can be interpreted as follows: the belief about a matter is influenced by a combination of looking at current evidences and the prior belief about the matter.

Mathematically, given a prior probability distribution on parameters $P(\theta)$ where θ refers to parameters, the likelihood function $P(X|\theta)$ where X correspond to observations, the posterior probability $P(\theta|X)$ (*i.e.* the distribution on the parameters after seeing the observations) can be written as:

$$P(\theta|X) = \frac{P(\theta)P(X|\theta)}{P(X)}. \tag{7}$$

Since the observations can be treated as a constant, $P(\theta|X)$ can also be formulated as:

$$P(\theta|X) \propto P(\theta) \times P(X|\theta). \tag{8}$$

This implies that:

$$\text{Posterior probability} \propto \text{Prior probability} \times \text{Likelihood} \tag{9}$$

In the context of this paper, the prior probability refers to the probability or belief about the bounding box location of an object *before* seeing any (weak) supervision. Furthermore, the posterior probability is the probability after seeing the weak supervision. It is to be noted that although our proposed algorithm can be interpreted in a Bayesian way, it is not strictly Bayesian.

Due to noises and inaccuracies in the background modelling and subtraction process *and* due to the fact that this prior has been learnt using all classes of foreground objects in the scene, the pedestrian prior is error prone and would give high probabilities not only for pedestrians but also other object categories such as vehicles.

However, we do not make any hard decisions at this stage and any errors and uncertainties in the pedestrian prior are resolved in the next stage using $\mathbb{S}_{\text{weak}}$.

The second stage involves an optimization framework with an objective function that is a mixture of two terms:

1. The score of the pedestrian prior obtained in the first stage. This can be interpreted as the prior probability in Equation 7.

2. The agreement with the centres $\mathbb{S}_{\text{weak}}$. This can be compared to the likelihood in Equation 7.

We perform the optimization independently for each centre $\mathbf{cc}_i \in \mathbb{S}_{\text{weak}}$. Formulating in this way is very efficient compared to having to solve them jointly. After optimizing each weak supervision annotation (described in Section 5.2), we automatically obtain a bounding box annotation corresponding to the weak supervision.

Therefore, the second stage is in essence automatically converting the set of weak centre annotations $\mathbb{S}_{\text{weak}}$ to a set of bounding box annotations which are represented by:

$$\mathbb{S}_{\text{bbox}} = \{\mathbf{bb}_1, \mathbf{bb}_2, \ldots, \mathbf{bb}_M\},$$

where

$$\mathbf{bb}_i = [bb_i^{x_1}, bb_i^{y_1}, bb_i^{x_2}, bb_i^{y_2}, bb_i^n]$$

is a 5-element vector with bb_i^n denoting the frame number in $\mathbb{V}$ associated with $\mathbf{bb}_i$, $bb_i^{x_1}$ and $bb_i^{y_1}$ giving the x and y coordinates (in image-plane) respectively of the upper left corner of the rectangle corresponding to the bounding box annotation and $bb_i^{x_2}$ and $bb_i^{y_2}$ representing the x and y coordinates of the bottom right corner of the bounding box.

After obtaining $\mathbb{S}_{\text{bbox}}$, we can now use any supervised learning algorithm to train a pedestrian detector. This is the third stage.

We formalise our approach in Algorithm 1 and explain it in detail in the coming sections. The inputs to Algorithm 1 are the video $\mathbb{V}$ and the weak supervision $\mathbb{S}_{\text{weak}}$ manually annotated by the user. The output is the desired pedestrian detector $\mathcal{C}_t$.

In this paper, we use the phrase "scene-specific pedestrian detector" to denote the fact that the pedestrian detector is trained using the data collected from the video $\mathbb{V}$ and although it will perform well for $\mathbb{V}$, we cannot make any guarantees on its generalisation power on other datasets. The term "scene-specific pedestrian detector" is used to distinguish it from other possible pedestrian detectors trained using "generic" datasets such as the INRIA Person dataset [3] which has been manually collected from various different types of data sources and whose goal is to make the trained detectors to be as generalisable as possible. In contrast, the goal of a scene-specific pedestrian detector such as $\mathcal{C}_t$ is relatively more modest and is to generalise only over $\mathbb{V}$.

Algorithm 1 Overview of the weakly supervised training

Input: Video $\mathbb{V}$ and weak supervision $\mathbb{S}_{\text{weak}}$
Output: Scene-specific pedestrian detector, $\mathcal{C}_t$

$\mathcal{C}_p \leftarrow$ LearnPrior($\mathbb{V}$), where LearnPrior is the function to learn unsupervised pedestrian prior. Described in Algorithm 2.

$\mathbb{S}_{\text{bbox}} \leftarrow$ FuseOptimize($\mathbb{V}, \mathbb{S}_{\text{weak}}, \mathcal{C}_p$), where FuseOptimize is the function to convert weak centre supervision $\mathbb{S}_{\text{weak}}$ to bounding box annotations $\mathbb{S}_{\text{bbox}}$. Detailed in Algorithm 3.

$\mathbf{X}_t^+ \leftarrow \mathcal{H}$(patches corresponding to $\mathbb{S}_{\text{bbox}}$)

$\mathbf{X}_t^- \leftarrow \varnothing$
for $I_i \in \mathbb{V}$ **do**
  Let $\mathbb{W}$ be the set of sliding windows on I_i
  $\mathbf{X}_t^- \leftarrow \mathbf{X}_t^- \cup \mathcal{H}(\{\mathbf{w} \in \mathbb{W} : \mathbf{w} \cap \mathbb{S}_{\text{bbox}} = \varnothing\})$
end for

$(\mathbf{X}_t^l, \mathbf{Y}_t) \leftarrow \{\mathbf{X}_t^+, \mathbf{X}_t^-\}$

$\mathcal{C}_t \leftarrow$ LearnClassifier($\mathbf{X}_t^l, \mathbf{Y}_t$)

return $\mathcal{C}_t$

5.1 Unsupervised Pedestrian Prior Learning

For a given video $\mathbb{V}$, a pedestrian prior, $\mathcal{C}_p : \mathbb{R}^{N_{\text{base}}} \rightarrow [0,1]$, is learnt in an unsupervised way, where N_{base} is the length of the feature vector input to the pedestrian prior and $\mathcal{C}_p$ outputs the (prior) probability that the given feature vector is a pedestrian.

Algorithm 2 Unsupervised pedestrian prior learning

Input: Video $\mathbb{V}$
Output: Unsupervised pedestrian prior $\mathcal{C}_p$

$\mathbf{X}_t^{\text{fg}} \leftarrow \varnothing$
$\mathbf{X}_t^{\text{bg}} \leftarrow \varnothing$

Let Ω be an initial estimate of the scene background.

for $I_i \in \mathbb{V}$ **do**

$\Omega \leftarrow \text{UpdateBGModel}(I_i, \Omega)$
$I_{\text{fgmask}} \leftarrow$ Background subtraction using $\{I_i, \Omega\}$
Connected Component Analysis on I_{fgmask}
$\mathbb{B} \leftarrow \{\text{bounding boxes of the connected blobs}\}$

for $\mathbf{b}_i \in \mathbb{B}$ **do**
$\mathbf{X}_t^{\text{fg}} \leftarrow \mathbf{X}_t^{\text{fg}} \cup \{\mathcal{H}(\mathbf{b}_i)\}$
end for

Let $\mathbb{W}$ be the set of sliding windows on I_i
$\mathbb{W} \leftarrow \{\mathbf{w} \in \mathbb{W} : \mathbf{w} \cap \mathbb{B} = \varnothing\}$
$\mathbf{X}_t^{\text{bg}} \leftarrow \mathbf{X}_t^{\text{bg}} \cup \mathcal{H}(\mathbb{W})$

end for

$(\hat{\mathbf{X}}_t^l, \hat{\mathbf{Y}}_t^l) \leftarrow \{\mathbf{X}_t^{\text{fg}}, \mathbf{X}_t^{\text{bg}}\}$
$\mathcal{C}_p \leftarrow \text{LearnClassifier}(\hat{\mathbf{X}}_t^l, \hat{\mathbf{Y}}_t^l)$
$\mathcal{C}_p \leftarrow$ Calibrate $\mathcal{C}_p$ to produce valid probabilities.

return $\mathcal{C}_p$

Let a feature extraction function be $\mathcal{H} : \mathbb{R}^{N_r \times N_c \times 3} \rightarrow \mathbb{R}^{N_{\text{base}}}$ where N_r and N_c are the number of rows and columns of an image patch and N_{base} is the length of the resulting feature vector.

Therefore, using $\mathcal{C}_p$ and $\mathcal{H}$, any image patch can be mapped to a value in the range of 0 and 1 (inclusive) that gives the prior probability that the given image patch is a pedestrian.

The method for learning $\mathcal{C}_p$ is outlined in Algorithm 2 and we describe the steps of the algorithm below.

Firstly, background subtraction is performed for each frame $I_i \in \mathbb{V}$, followed by Connected Component Analysis (CCA). CCA (sometimes also called Connected Component Labelling or blob discovery) is used to assign, by using a heuristic, each white (i.e. foreground) pixel in a binary image to a particular group based on its spatial proxmity to other foreground pixels. This has the effect of discovering foreground blobs in the binary image. For each discovered blob, the tightest fitting axis-aligned rectangle (i.e. bounding box) can be found. Therefore, CCA gives a set of the bounding boxes corresponding to the connected components which are stored in a set $\mathbb{B}$.

All background modelling techniques make use the fact that for videos captured with a static camera, background can be defined as areas in the scene that do not change much over a period of time and that are relatively stable in terms of appearance as compared to foreground areas. These background regions can be statistically modelled and from the background model, foreground regions can be deduced.

There are various ways of modelling the background. Two main approaches are: (1) modelling each pixel independently and then post-processing the results using some spatial information and (2) modelling groups of pixels with no dependence among each group. There have been many different statistical models developed for modelling the background; some examples are median filter [30], linear predictive filter [31], Gaussian Mixture Models [32], non-parametric models [33] and Dirichlet Process Mixture Models [34].

In this paper, we do not go into detail about background subtraction; for a review on background subtraction, the reader is referred to various surveys such as [35, 36]. Any suitable background subtraction technique could be used with our proposed weakly supervised learning algorithm. However, since the unsupervised prior learning stage is offline and does not need real-time processing, a highly accurate and robust yet reasonably fast background subtraction algorithm (such as [37]) is recommended and is used in this paper.

For each underlying image patch corresponding to a bounding box $\mathbf{b}_i \in \mathbb{B}$, we compute features by applying the function $\mathcal{H}$ (after appropriate resizing of the patch). These feature vectors form a set $\mathbf{X}_t^{\text{fg}}$. The feature extraction function is general and any suitable method can be used. In this paper, we use the well-known Histograms of Oriented Gradients (HOGs) features [3].

Random samples of patches from which $\mathbf{X}_t^{\text{fg}}$ is obtained are shown in Figures 4 and 5 for the CUHK Square [14] and MIT Traffic [38] datasets respectively.

Now, we take all the multi-scale sliding windows in each frame $I_i \in \mathbb{V}$ that do *not* overlap with any $\mathbf{b}_i \in \mathbb{B}$, resize and extract features using $\mathcal{H}$, giving the set of feature vectors $\mathbf{X}_t^{\text{bg}}$.

Finally, since we have collected both $\mathbf{X}_t^{\text{fg}}$ and $\mathbf{X}_t^{\text{bg}}$, we train a binary classifier $\mathcal{C}_p$ using $\mathbf{X}_t^{\text{fg}}$ as the positive class and $\mathbf{X}_t^{\text{bg}}$ as the negative class. This classifier (*i.e.* function) $\mathcal{C}_p$ is the pedestrian prior.

The aim of $\mathcal{C}_p$ is simply to capture some information about the pedestrian class. $\mathcal{C}_p$ *implicitly* captures the multi-modal distribution about objects in the scene and obtaining $\mathcal{C}_p$ therefore does not require clustering and explictly discovering foreground object categories. $\mathcal{C}_p$ could be any type of classifier and in our case, a linear SVM [39] is used.

If $\mathcal{C}_p$ does not output valid probabilities (such as when using an SVM), $\mathcal{C}_p$ should be calibrated to produce (proper) probabilities. One such way to calibrate it by using Platt scaling [40]. It is essentially fitting a logistic regression on the classifier scores:

$$\mathcal{C}_p(\mathbf{x}) \leftarrow \frac{1}{1+\mathrm{e}^{-(\beta_0+\beta_1\mathcal{C}_p(\mathbf{x}))}}, \tag{10}$$

where β_0 and β_1 are scalar parameters associated with the logistic regression and can be learnt from data and $\mathbf{x}$ is the feature vector of a data point that is input to the $\mathcal{C}_p$.

5.2 Cue Fusion and Optimization

The goal here is to fuse the unsupervised pedestrian prior $\mathcal{C}_p$ obtained (as described in the previous section) with the provided weak supervisions $\mathbb{S}_{\text{weak}}$ and optimize the combination of these two sources of information to generate bounding box annotations $\mathbb{S}_{\text{bbox}}$. The technique is formally given in Algorithm 3 and explained below.

The set of generated bounding box annotations $\mathbb{S}_{\text{bbox}}$ is initialised to an empty set $\varnothing$. There are M weak supervisions (*i.e.* $|\mathbb{S}_{\text{weak}}| = M$) and each weak supervision $\mathbf{cc}_i \in \mathbb{S}_{\text{weak}}$ is optimized independently. Therefore, below, we detail the process of converting a *single* weak supervision $\mathbf{cc}_i \in \mathbb{S}_{\text{weak}}$ to a single bounding box annotation $\mathbf{bb}_i \in \mathbb{S}_{\text{bbox}}$.

Firstly, we compute a large rectangular region Φ surrounding and centred at the weak supervision $[cc_i^x, cc_i^y]$. This can be computed by setting for the whole video, *estimates* of the widths and heights of the smallest and largest possible pedestrian in the scene. These do not need to be accurate, can be easily determined by a human and need to be set only once at the beginning of the algorithm.

Then, a set of K multi-scale sliding windows

$$\mathbb{W} = \{\mathbf{w}_1, \ldots, \mathbf{w}_K\}$$

Algorithm 3 Cue Fusion and Optimization

Input: Video $\mathbb{V}$, weak supervision $\mathbb{S}_{\text{weak}}$ and unsupervised pedestrian prior $\mathcal{C}_p$
Output: Bounding box annotations $\mathbb{S}_{\text{bbox}}$

$\mathbb{S}_{\text{bbox}} \leftarrow \varnothing$

Let $\{w_{\min}, w_{\max}, h_{\min}, h_{\max}\}$ be estimates of minimum and maximum possible widths and heights of pedestrians in $\mathbb{V}$.

for $i = 1$ **to** M **do**

% Get info from the current weak supervision $\mathbf{cc}_i \in \mathbb{S}_{\text{weak}}$ %
$\mathbf{cc}_i = [cc_i^x, cc_i^y, cc_i^n]$

% Get a large enough rectangular region surrounding current weak supervision %
$\Phi \leftarrow [cc_i^x - w_{\max}/2, cc_i^y - h_{\max}/2, cc_i^x + w_{\max}/2, cc_i^y + h_{\max}/2]$

$\mathbb{W} \leftarrow$ get multiscale sliding windows, larger than $w_{\min}$ and $h_{\min}$, in the region Φ. $\mathbb{W}$ is a set of K bounding boxes specified by $\{\mathbf{w}_1, \ldots, \mathbf{w}_K\}$ where $\mathbf{w}_j = [w_j^{x_1}, w_j^{y_1}, w_j^{x_2}, w_j^{y_2}]$ is a vector denoting the x and y coordinates of the top-left ($w_j^{x_1}$ and $w_j^{y_1}$) and the bottom-right ($w_j^{x_2}$ and $w_j^{y_2}$) corners of the bounding box $\mathbf{w}_j$.

Let $\mathcal{N}(\mathbf{w})$ be a function, $\mathcal{N} : \mathbb{R}^4 \rightarrow \mathbb{R}$, given by: $\mathcal{N}(\mathbf{w}) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}\left(\mathcal{G}(\mathbf{w}) - \mu\right)^T \Sigma^{-1} \left(\mathcal{G}(\mathbf{w}) - \mu\right)\right)$ where $\mu = [cc_i^x, cc_i^y]$, $\Sigma = \left[\begin{smallmatrix} 3 & 0 \\ 0 & 3 \end{smallmatrix}\right]$ and $\mathcal{G}(\mathbf{w}) = \left[\frac{w^{x_1} + w^{x_2}}{2}, \frac{w^{y_1} + w^{y_2}}{2}\right]$.

$$\hat{\mathbf{w}} = \arg\max_{\mathbf{w} \in \mathbb{W}} \frac{\mathcal{C}_p(\mathcal{H}(\mathbf{w}))}{\sum_{\mathbf{w} \in \mathbb{W}} \mathcal{C}_p(\mathcal{H}(\mathbf{w}))} + \frac{\mathcal{N}(\mathbf{w})}{\sum_{\mathbf{w} \in \mathbb{W}} \mathcal{N}(\mathbf{w})}$$

$\mathbf{bb}_i \leftarrow [\hat{w}^{x_1}, \hat{w}^{y_1}, \hat{w}^{x_2}, \hat{w}^{y_2}, cc_i^n]$ where $\mathbf{bb}_i \in \mathbb{S}_{\text{bbox}}$

end for

return $\mathbb{S}_{\text{bbox}}$

Figure 4: CUHK Square dataset [14]: Random samples of patches corresponding to foreground objects which form $\mathbf{X}_t^{\text{fg}}$ (after feature extraction). Generalising (*i.e.* learning) over these examples serves as a *pedestrian prior* for the CUHK Square scene.

are generated within Φ (and only windows whose widths and heights are larger than the smallest width and height set previously are retained). Each $\mathbf{w}_j \in \mathbb{W}$ is a bounding box and is defined by a vector

$$\mathbf{w}_j = [w_j^{x_1}, w_j^{y_1}, w_j^{x_2}, w_j^{y_2}],$$

where $w_j^{x_1}$ and $w_j^{y_1}$ are the x and y coordinates of the top-left corner of the bounding box and $w_j^{x_2}$ and $w_j^{y_2}$ are the x and y coordinates of the bottom-right corner of the bounding box.

After generating $\mathbb{W}$, we are now ready to fuse the unsupervised pedestrian prior $\mathcal{C}_p$ with the weak supervision to obtain a bounding box annotation. This is done in an optimization framework for which the objective function is essentially a combination of two terms:

1. The appearance term provided by the pedestrian prior $\mathcal{C}_p$: for a candidate window $\mathbf{w} \in \mathbb{W}$, the appearance term determines how likely the image patch corresponding to $\mathbf{w}$ is a pedestrian.

2. The spatial-scoring term provided by the weak supervision: for a candidate window $\mathbf{w} \in \mathbb{W}$, the spatial term gives the closeness of the centre of $\mathbf{w}$ to the weak supervision centre $[cc_i^x, cc_i^y]$.

Figure 5: MIT Traffic dataset [38]: Random samples of patches corresponding to foreground objects which form $\mathbf{X}_t^{\text{fg}}$. Generalising (*i.e.* learning) over these examples serves as a *pedestrian prior* for the MIT traffic scene.

We seek the best window $\hat{\mathbf{w}} \in \mathbb{W}$ such that $\hat{\mathbf{w}}$ is scored highest by the combination of these terms in the objective function as given below:

$$\hat{\mathbf{w}} = \arg\max_{\mathbf{w}\in\mathbb{W}} \frac{\mathcal{C}_p(\mathcal{H}(\mathbf{w}))}{\sum_{\mathbf{w}\in\mathbb{W}} \mathcal{C}_p(\mathcal{H}(\mathbf{w}))} + \frac{\mathcal{N}(\mathbf{w})}{\sum_{\mathbf{w}\in\mathbb{W}} \mathcal{N}(\mathbf{w})}. \tag{11}$$

The function $\mathcal{N} : \mathbb{R}^4 \rightarrow \mathbb{R}$, gives the likelihood of a window $\mathbf{w}$ to be consistent with the weak supervision and is defined as:

$$\mathcal{N}(\mathbf{w}) = \frac{1}{(2\pi)^{n/2}\,|\Sigma|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}\left(\mathcal{G}(\mathbf{w})-\mu\right)^T \Sigma^{-1}\left(\mathcal{G}(\mathbf{w})-\mu\right)\right), \tag{12}$$

where

$$\mu = [cc_i^x, cc_i^y],$$

$$\Sigma = \begin{bmatrix} 3 & 0 \\ 0 & 3 \end{bmatrix},$$

and

$$\mathcal{G}(\mathbf{w}) = \left[\frac{w^{x_1} + w^{x_2}}{2}, \frac{w^{y_1} + w^{y_2}}{2}\right].$$

$\mathcal{G}$ is a function, $\mathcal{G} : \mathbb{R}^4 \rightarrow \mathbb{R}^2$, to get the centre coordinates of a bounding box $\mathbf{w}$.

The terms in the denominator of the optimization function are normalisation terms to make sure the relative weighing of the two terms in the objective function are equal.

Informally, the optimization objective prefers $\mathbf{w} \in \mathbb{W}$ that is scored highly by $\mathcal{C}_p$ but is penalised the further the $\mathbf{w}$ is from the weak supervision centre $[cc_i^x, cc_i^x]$.

The function $\mathcal{N}$ is actually a bivariate Gaussian distribution with the mean μ at the weak supervision centre and the covariance Σ is fixed a priori. The function allows for uncertainty and noise in the weak supervision annotation process.

If the values in the covariance matrix Σ are too large, then the Gaussian weighing function would be too flat and it would fail to penalise bounding boxes that are far from $[cc_i^x, cc_i^y]$. In contrast, if the values are too small, then only the bounding boxes who centres are almost exactly the same as the centre supervision would be allowed. This would then result in suboptimal results when the centre supervision is noisy. The Σ is fixed to:

$$\Sigma = \begin{bmatrix} 3 & 0 \\ 0 & 3 \end{bmatrix}$$

before starting any experiments since it seems to be a set of reasonable values. The Gaussian function should penalise the same in any direction which is why a spherical covariance is used. We have not changed or manually tuned these values for any of the experiments, yet it turns out that this simple spherical covariance gives good results. Visualisation of this spatial penalty term is illustrated in Figures 6 and 7.

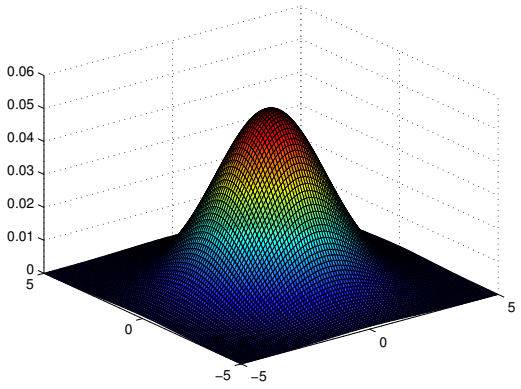


Figure 6: Bivariate normal distribution with mean $\mu = [0,0]$ and diagonal covariance matrix $\Sigma = \left[\begin{smallmatrix} 3 & 0 \\ 0 & 3 \end{smallmatrix}\right]$.

(a) (b) (c) (d) (e) (f)

Figure 7: Visualisation of 100 samples from the bivariate normal distribution with mean $\vec{\mu}$ at the weak supervision centre and diagonal covariance matrix $\Sigma = \left[\begin{smallmatrix} 3 & 0 \\ 0 & 3 \end{smallmatrix}\right]$ overlapped on the snippets of frames. Pedestrians are cropped from the MIT Traffic dataset [38].

5.3 Training the Scene-specific Detector

At the end of the cue fusion and optimization discussed in Section 5.2, a set of bounding box annotations, $\mathbb{S}_{\text{bbox}}$, for the target video, $\mathbb{V}$, is obtained.

In order to get the scene-specific positive data $\mathbf{X}_t^+$, the patches corresponding to $\mathbb{S}_{\text{bbox}}$ are cropped out and feature extraction function is run on each of the patches (after appropriate resizing). These feature vectors make up $\mathbf{X}_t^+$.

The scene-specific negative data $\mathbf{X}_t^-$ are obtained in a similar fashion except that patches are cropped out from regions that do not intersect with $\mathbb{S}_{\text{bbox}}$.

$\mathbf{X}_t^+$ and $\mathbf{X}_t^-$ give a (strongly) labelled target dataset $(\mathbf{X}_t^l, \mathbf{Y}_t)$ which can be used to train a scene-specific pedestrian detector $\mathcal{C}_t$. Any type of classifier can be used. We (again) use linear SVM for simplicity. Not only the classifier type but the feature extraction function is also general and the function used in this stage does not have to the same as the one used during the unsupervised prior learning.

6 Results and Discussion

We have used the challenging CUHK Square [14] and MIT Traffic [38] video datasets to evaluate our weakly supervised learning algorithm. These videos have been obtained from a static camera that records a far-field traffic scene that consists of various object categories such as pedestrians, cars, buses and cyclists. They are also very challenging in terms of the small resolution of pedestrians, illumination variations, *etc*.

The CUHK Square dataset contains a total of 90,425 frames (corresponding to about 60 minutes of video recorded with 25 fps) and each frame is of size 720×576. In the MIT Traffic dataset, there are 165,880 frames (approximately 90 minutes of video at 30 fps) with the frame size of 720×480.

We note that the goal of our research is *not* about obtaining the absolute maximum performance in pedestrian detection (which is important in its own right), but in weakly supervised learning. The performance of a pedestrian detector is directly related to the feature extraction process, the type of classifier, the non-maximum suppression amongst other factors. In this paper, factors such as feature extraction and classifier type are fixed constant so that a meaningful comparison is made for evaluating the weakly supervised learning algorithm. Moreover, our proposed system can be used to train any state-of-the-art pedestrian detector (with a given feature extraction and classification algorithm).

The upper bound (*i.e*. the maximum possible) performance for any weakly supervised learning algorithm is the performance obtained with when all the pedestrian bounding boxes are manually annotated by humans in the training dataset. This is because it can be assumed that the best bounding box locations are those provided by humans. Therefore, in our experiments, we directly compare our approach with this upper bound.

For each video dataset, we split it to two equal halves. During weakly-supervised training (including unsupervised prior learning), we only use the first half (called the "training dataset"). The second half (*i.e*. the "test dataset") is kept purely for evaluating the resulting pedestrian detectors. This is to ensure that during training, the system does not get any glimpse of the test data.

Ground truth pedestrian bounding boxes are available for both the training and test datasets. However, the ground truth bounding boxes are not used for training the pedestrian detector using our proposed weakly supervised learning algorithm. They are used only for training the aforementioned upper bound pedestrian detector. For training, there are a total of 982 supervisions given for the CUHK Square dataset and 1573 for the MIT Traffic dataset. We set the hyperparameters $\{w_{\min}, w_{\max}, h_{\min}, h_{\max}\}$ (mentioned in Algorithm 3) for the CUHK Square dataset to be $\{5, 100, 10, 200\}$ and for the MIT Traffic dataset, we set them as $\{5, 50, 10, 100\}$.

After training the pedestrian detector on the training dataset, the detector is run on the test data to get pedestrian detections. PASCAL 50% overlap criterion [41] is used to decide whether the detected bounding boxes are correct with respect to the ground truth bounding boxes. Given the ground truth bounding box $\mathbf{b}_g$ and the detected bounding box $\mathbf{b}_d$, the overlap, α_o, gives the overlap proportion between $\mathbf{b}_g$ and $\mathbf{b}_d$ and is defined as follows:

$$\alpha_o = \frac{\text{area}(\mathbf{b}_g \cap \mathbf{b}_d)}{\text{area}(\mathbf{b}_g \cup \mathbf{b}_d)}. \tag{13}$$

According to PASCAL 50% overlap criterion, if $\alpha_o > 0.5$, a detection is marked as correct.

For evaluation and performance comparison, Recall-FPPI (False Positives Per Image) curves are used since they are commonly used in state-of-the-art object detection research [42, 14]. Especially for pedestrian detection, Recall-FPPI curves provides information, at a glance, about the recall at various false positives per image.

We perform three different types of experiments on each dataset:

1. The pedestrian detector obtained by our weakly supervised algorithm.
2. The detector obtained by "strong" supervision (manual bounding box annotation). This will indicate the upper bound performance. A hundred uniformly sampled frames annotated with ground truth bounding boxes from the training dataset are used for training.
3. The detector corresponding to the unsupervised prior (as described in Algorithm 2).

These experiments are named *Weak supervision*, *Strong supervision* and *Unsupervised prior* respectively in the curves shown in Figure 8. In addition, Figure 9 illustrates the cost comparison between the weak and strong supervisions.

As can be seen, the detection performance of the proposed algorithm closely matches that of the strong supervision (*i.e.* the upper bound). Yet, the time it took to manually annotate training data for the proposed algorithm was less than one quarter of the time taken for the strong supervision.

This means that our algorithm reduces the manual human annotation effort by over four times to get the same performance as the standard strongly supervised training in literature.

We also evaluated unsupervised prior in order to show the effectiveness of our fusion and optimization framework. The unsupervised prior alone performs poorly; however, when fused with the weak supervision, the resulting detector has a much higher performance than the unsupervised prior. This means that in terms of Bayesian machine learning as outlined in Equation 7 (Section 5), the prior successfully captures sufficient probabilities in the right regions. Additionally, the likelihood function is a suitable one since it places the right weights on the prior distribution so that the max of the posterior distribution results in the correct weakly supervised learning.

The reason for the proposed algorithm performing very slightly lower than the strong supervision may be because of some minor inaccuracies in the generated bounding boxes (the output of the cue fusion and optimization), which result in very small bounding box misalignments (to the actual pedestrians) compared to strong supervision (where the bounding boxes are manually given by humans resulting in less such alignment errors).

For completeness, rather than evaluating the entire system as a whole as done previously, we also show the evaluation of the generated bounding boxes of the cue fusion and optimization step (*i.e.* the output of Algorithm 3) by directly comparing with the ground truth strong supervision (bounding boxes). This is useful for getting an understanding of how close the generated bounding boxes are to the ground truth bounding boxes.

To this end, we compute the overlap α_o (defined in Equation 13) of each generated bounding box with the corresponding ground truth bounding box. And then, we take the median of these overlap scores across all the generated bounding boxes. We term this as the *median overlap score*. We also perform a baseline experiment by removing the spatial penalty term (*i.e.* setting the spatial penalty term set to zero). The results are shown in Table 1 and Figure 10, and they verify the results that we have obtained and discussed previously: the proposed fusion and optimization is highly effective and the generated bounding boxes are very close to the ground truth bounding boxes (as shown by the median overlap scores of 0.8902 and 0.8570 for the CUHK Square and MIT Traffic datasets respectively). Moreover, only using the pedestrian prior without the proposed fusion of the

	No spatial penalty term	Fused System
CUHK Square	0.2032	0.8902
MIT Traffic	0.1854	0.8570
Mean	0.1943	0.8736

Table 1: Comparison of median overlap scores.

pedestrian prior with the weak supervision labels (*i.e.* approximate object center annotations), the generated bounding boxes are found to be very poor (resulting in the median overlap scores of 0.2032 and 0.1854 for the two datasets). These observations correlate well with the results established previously in this section.

7 Conclusions and Future Work

We have proposed a novel weakly supervised learning algorithm for training pedestrian detectors for videos. The algorithm consists of learning an unsupervised prior using unlabelled data in the video and then fusing the prior with the weak supervision in an optimization framework to generate bounding box annotations. We have shown that the weakly supervised algorithm can reduce the amount of human annotation effort by over four times without sacrificing the accuracy of the resulting detector.

Possible directions for future research include investigating a weakly supervised learning approach where the annotation is simply marking anywhere on (or even near) the pedestrian's body rather than the approximate centre location. This should allow for much faster annotation speed. With this approach, however, it may be necessary to perform (possibly expensive) joint optimization of all the weak supervision annotations (exploiting the appearance similarity between pedestrians) rather than independent optimization for each weak supervision as afforded by algorithm proposed in this paper.

Another direction is to make the covariance of the spatial penalty term to somehow depend on the (unknown) scale of the pedestrian so that the system properly handle noise in the annotated centre location for a wide range of pedestrian scales. Furthermore, there could be alternative fusion and optimization formulations constructed based on the idea in this paper; one possible approach is using a fully Bayesian approach to define priors and likelihoods, and then estimating the posterior distribution in order to generate the pedestrian bounding boxes from the weak supervisions.

Moreover, it will also be an interesting direction to explore a more rigorous and large-scale human-computer interaction experiments to study the expected cost saving from various weakly supervised learning approaches.

Finally, it may be useful to investigate in the future to combine, in a principled way, weakly supervised learning, semi-supervised learning and active learning to achieve the minimum annotation effort required for labelling large datasets.

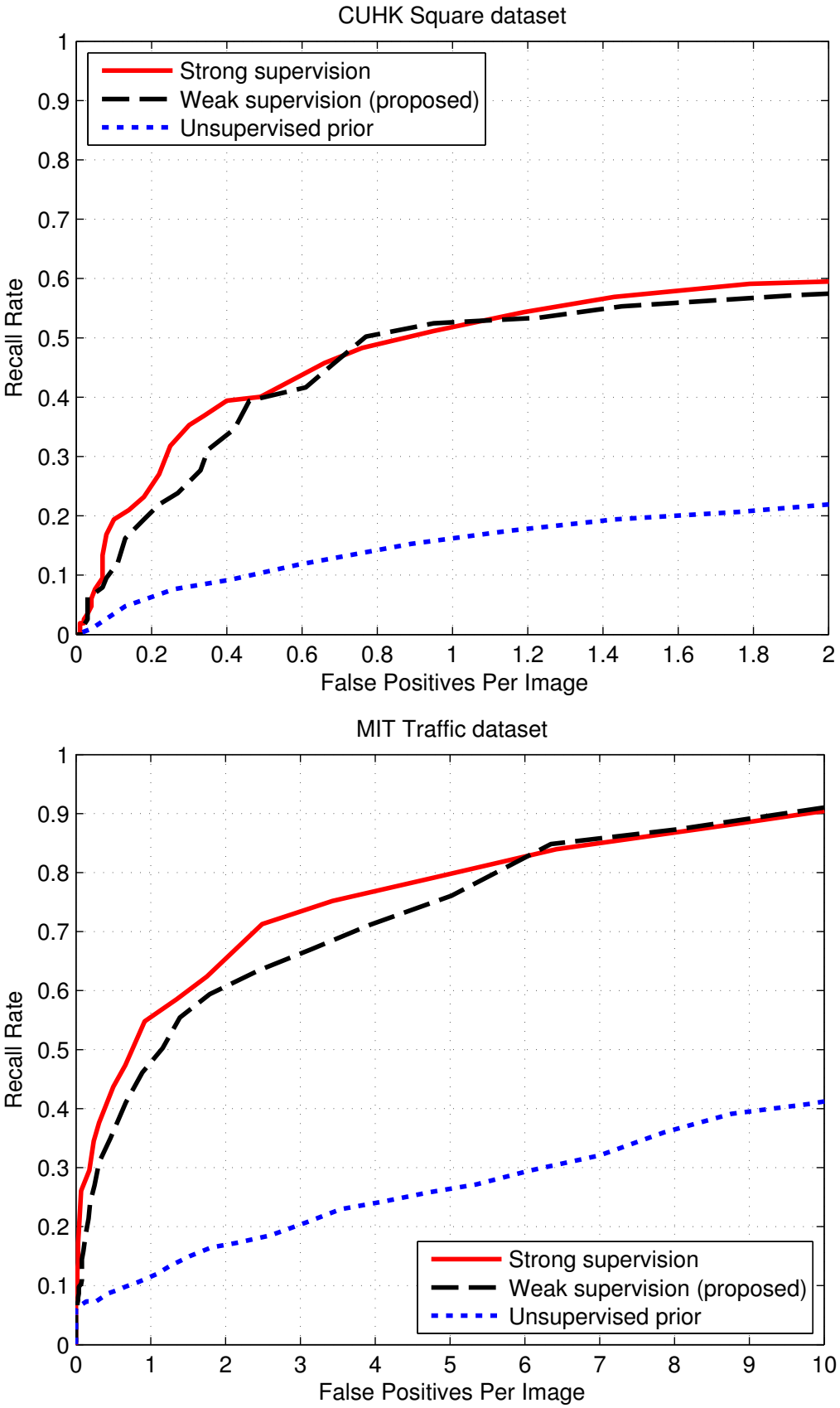


Figure 8: Detection performance curves for CUHK (top) and MIT (bottom)

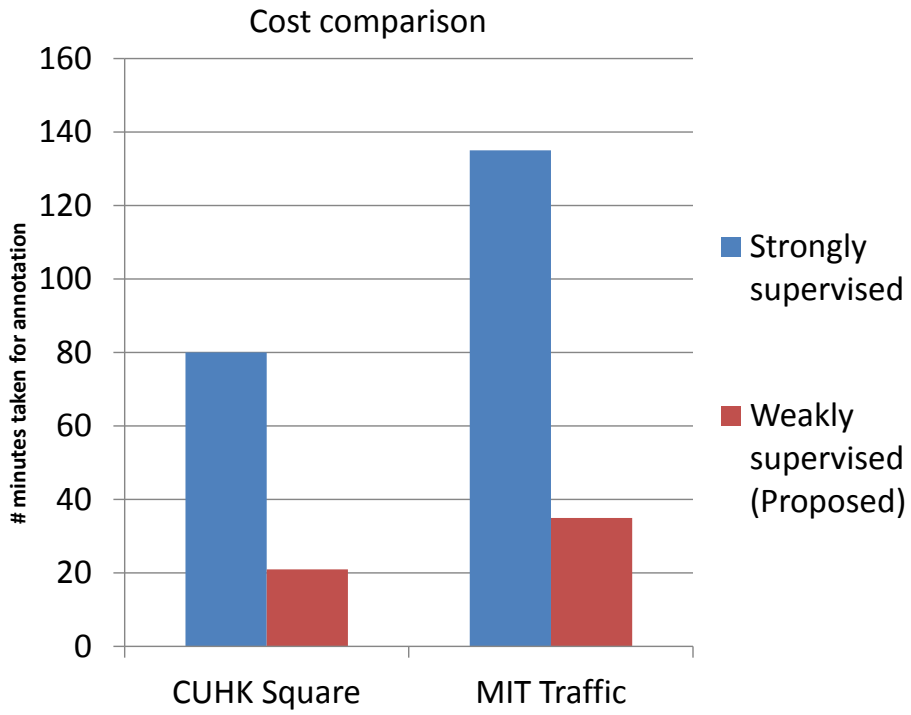


Figure 9: Cost comparison

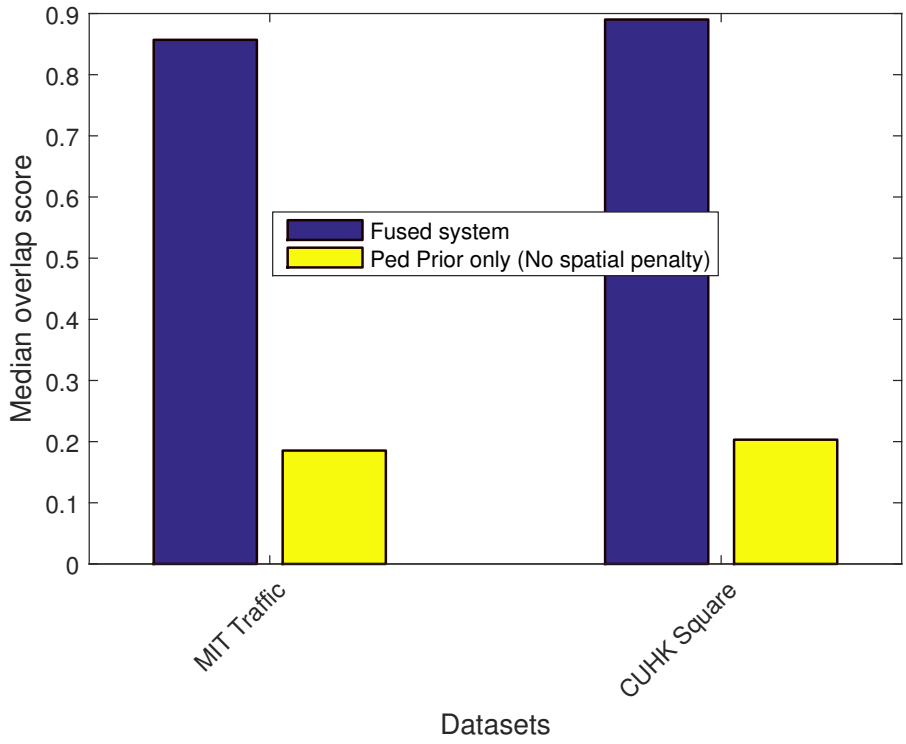


Figure 10: Median overlap scores

References

[1] Constantine Papageorgiou and Tomaso Poggio. A trainable system for object detection. *International Journal of Computer Vision*, 38(1):15–33, 2000. doi: 10.1023/A:1008162616689.

[2] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages I–511. IEEE, 2001. doi: 10.1109/cvpr.2001.990517.

[3] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 886–893, 2005. doi: 10.1109/cvpr.2005.177.

[4] Kyaw Kyaw Htike and David Hogg. Unsupervised detector adaptation by joint dataset feature learning. In *Computer Vision and Graphics*, pages 270–277. Springer, 2014. doi: 10.1007/978-3-319-11331-9_33.

[5] Kyaw Kyaw Htike and David Hogg. Efficient non-iterative domain adaptation of pedestrian detectors to video scenes. In *International Conference on Pattern Recognition (ICPR)*, pages 654–659. IEEE, 2014. doi: 10.1109/icpr.2014.123.

[6] Piotr Dollár, Christian Wojek, Bernt Schiele, and Pietro Perona. Pedestrian detection: An evaluation of the state of the art. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(4):743–761, 2012. doi: 10.1109/tpami.2011.155.

[7] Markus Enzweiler and Dariu M. Gavrila. Monocular pedestrian detection: Survey and experiments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(12):2179–2195, 2009. doi: 10.1109/tpami.2008.260.

[8] Ross B Girshick, Pedro F Felzenszwalb, and David A Mcallester. Object detection with grammar models. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, pages 442–450, 2011.

[9] Piotr Dollár, Ron Appel, Serge Belongie, and Pietro Perona. Fast feature pyramids for object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2014. doi: 10.1109/tpami.2014.2300479.

[10] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645, 2010. doi: 10.1109/tpami.2009.167.

[11] P. Dollár, R. Appel, and W. Kienzle. Crosstalk cascades for frame-rate pedestrian detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 645–659, 2012. doi: 10.1007/978-3-642-33709-3_46.

[12] Rodrigo Benenson, Markus Mathias, Radu Timofte, and Luc Van Gool. Pedestrian detection at 100 frames per second. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2903–2910. IEEE, 2012. doi: 10.1109/cvpr.2012.6248017.

[13] Marco Pedersoli, Andrea Vedaldi, and Jordi Gonzalez. A coarse-to-fine approach for fast deformable object detection. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1353–1360. IEEE, 2011. doi: 10.1109/cvpr.2011.5995668.

[14] Meng Wang, Wei Li, and Xiaogang Wang. Transferring a generic pedestrian detector towards specific scenes. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 3274–3281, 2012. doi: 10.1109/cvpr.2012.6248064.

[15] Thomas G Dietterich, Richard H Lathrop, and Tomás Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence*, 89(1):31–71, 1997. doi: 10.1016/s0004-3702(96)00034-3.

[16] Stuart Andrews, Ioannis Tsochantaridis, and Thomas Hofmann. Support Vector Machines for Multiple-instance Learning. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, pages 561–568, 2002.

[17] Carolina Galleguillos, Boris Babenko, Andrew Rabinovich, and Serge J. Belongie. Weakly supervised object localization with stable segmentations. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 193–207, 2008. doi: 10.1007/978-3-540-88682-2_16.

[18] Yixin Chen and James Z Wang. Image categorization by learning and reasoning with regions. *The Journal of Machine Learning Research*, 5:913–939, 2004.

[19] Ondrej Chum and Andrew Zisserman. An exemplar model for learning object classes. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8. IEEE, 2007. doi: 10.1109/cvpr.2007.383050.

[20] Minh Hoai Nguyen, Lorenzo Torresani, Fernando de la Torre, and Carsten Rother. Weakly supervised discriminative localization and classification: a joint learning process. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 1925–1932, 2009. doi: 10.1109/iccv.2009.5459426.

[21] Andreas Opelt and Axel Pinz. Object localization with boosting and weak supervision for generic object recognition. In *Image Analysis*, pages 862–871. Springer, 2005. doi: 10.1007/11499145_87.

[22] Thomas Deselaers, Bogdan Alexe, and Vittorio Ferrari. Localizing objects while learning their appearance. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 452–466, 2010. doi: 10.1007/978-3-642-15561-1_33.

[23] Matthew Blaschko, Andrea Vedaldi, and Andrew Zisserman. Simultaneous object detection and ranking with weak supervision. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, pages 235–243, 2010.

[24] Alessandro Prest, Christian Leistner, Javier Civera, Cordelia Schmid, and Vittorio Ferrari. Learning object class detectors from weakly annotated video. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 3282–3289, 2012. doi: 10.1109/cvpr.2012.6248065.

[25] Hakan Bilen, Marco Pedersoli, and Tinne Tuytelaars. Weakly supervised object detection with posterior regularization. In *British Machine Vision Conference*, 2014. doi: 10.5244/c.28.52.

[26] Hakan Bilen, Marco Pedersoli, and Tinne Tuytelaars. Weakly supervised object detection with convex clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1081–1089, 2015. doi: 10.1109/cvpr.2015.7298711.

[27] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[28] Yann LeCun and Yoshua Bengio. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10), 1995.

[29] Maxime Oquab, Léon Bottou, Ivan Laptev, and Josef Sivic. Is object localization for free?–weakly-supervised learning with convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 685–694, 2015. doi: 10.1109/cvpr.2015.7298668.

[30] Rita Cucchiara, Costantino Grana, Massimo Piccardi, and Andrea Prati. Detecting moving objects, ghosts, and shadows in video streams. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(10): 1337–1342, 2003. doi: 10.1109/tpami.2003.1233909.

[31] Kentaro Toyama, John Krumm, Barry Brumitt, and Brian Meyers. Wallflower: Principles and practice of background maintenance. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 1, pages 255–261. IEEE, 1999. doi: 10.1109/iccv.1999.791228.

[32] Nir Friedman and Stuart Russell. Image segmentation in video sequences: A probabilistic approach. In *Proceedings of the Thirteenth conference on Uncertainty in artificial intelligence*, pages 175–181. Morgan Kaufmann Publishers Inc., 1997.

[33] Ahmed Elgammal, David Harwood, and Larry Davis. Non-parametric model for background subtraction. In *ECCV*, pages 751–767. Springer, 2000. doi: 10.1007/3-540-45053-x_48.

[34] Tom SF Haines and Tao Xiang. Background subtraction with dirichletprocess mixture models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 36(4):670–683, 2014. doi: 10.1109/tpami.2013.239.

[35] Massimo Piccardi. Background subtraction techniques: a review. In *IEEE International Conference on Systems, Man and Cybernetics (SMC)*, pages 3099–3104, 2004. doi: 10.1109/icsmc.2004.1400815.

[36] Sen-Ching S Cheung and Chandrika Kamath. Robust techniques for background subtraction in urban traffic video. In *Proceedings of SPIE*, volume 5308, pages 881–892, 2004. doi: 10.1117/12.526886.

[37] Jian Yao and Jean-Marc Odobez. Multi-layer background subtraction based on color and texture. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8. IEEE, 2007. doi: 10.1109/cvpr.2007.383497.

[38] Meng Wang and Xiaogang 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)*, pages 3401–3408, 2011. doi: 10.1109/cvpr.2011.5995698.

[39] Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152. ACM, 1992. doi: 10.1145/130385.130401.

[40] John Platt et al. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74, 1999.

[41] Mark Everingham, Luc J. Van Gool, Christopher K. I. Williams, John M. Winn, and Andrew Zisserman. The Pascal Visual Object Classes (VOC) challenge. *International Journal of Computer Vision*, 88(2): 303–338, 2010. doi: 10.1007/s11263-009-0275-4.

[42] Piotr Dollár, Christian Wojek, Bernt Schiele, and Pietro Perona. Pedestrian detection: A benchmark. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 304–311, 2009. doi: 10.1109/cvpr.2009.5206631.