Efficient Non-iterative Domain Adaptation of Pedestrian Detectors to Video Scenes

Kyaw Kyaw Htike
School of Computing
University of Leeds
Leeds, UK
sckkh@leeds.ac.uk

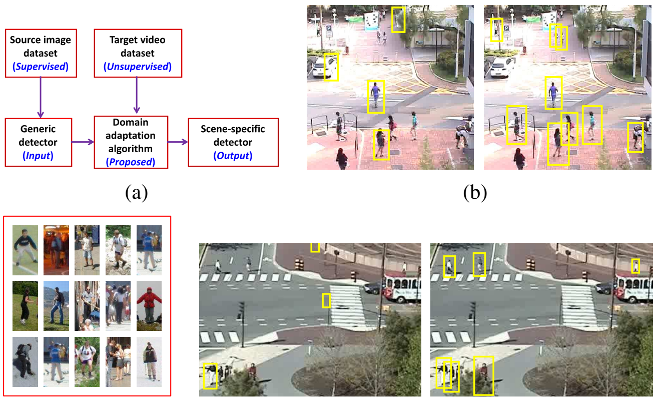
David Hogg
School of Computing
University of Leeds
Leeds, UK
d.c.hogg@leeds.ac.uk

***Abstract*—Pedestrian detection is an essential step in many important applications of Computer Vision. Most detectors require manually annotated ground-truth to train, the collection of which is labor intensive and time-consuming. Generally, this training data is from representative views of pedestrians captured from a variety of scenes. Unsurprisingly, the performance of a detector on a new scene can be improved by tailoring the detector to the specific viewpoint, background and imaging conditions of the scene. Unfortunately, for many applications it is not practical to acquire this scene-specific training data by hand. In this paper, we propose a novel algorithm to automatically adapt and tune a generic pedestrian detector to specific scenes which may possess different data distributions than the original dataset from which the detector was trained. Most state-of-the-art approaches can be inefficient, require manually set number of iterations to converge and some form of human intervention. Our algorithm is a step towards overcoming these problems and although simple to implement, our algorithm exceeds state-of-the-art performance.**

I. Introduction

Pedestrian detection in monocular images is a challenging task and a lot of progress has been made in this area (see [1], [2] for review and comparisons). The main approach to train a pedestrian detector is to get a *generic* dataset large enough to capture as many intra-class variations of pedestrians as possible. However, no dataset can possibly capture all the possible variations the detector is likely to face in the real world and therefore the detector may fail to perform satisfactorily when applied to scenes which have different data distributions than the generic dataset [3], [4]. This can be solved by training *scene-specific pedestrian detectors* which are tuned to specific scenes but this requires collecting labelled data in every new scene encountered and training a new detector which can be labor intensive.

The goal of this paper is shown in Fig. 1. We have a *source* dataset with supervision given in the form of pedestrian annotations and a *target* video dataset where supervision information is *not* available. The pedestrians in the target video have a different data distribution than the ones in the source dataset due to factors such as different poses, image resolution, camera angle and illumination variations. The source dataset is a generic (annotated) pedestrian dataset which is publicly available. The aim is to *automatically* generate a *scene-specific* detector which is tuned to the target video and would therefore perform better than the generic detector. This may seem like an infeasible task since no supervision from the target video is available. However, this is not usually the case because there are certain assumptions we can make about the structure of the underlying distribution of the source data and target data [5]. This is a *domain adaptation* problem.


Source image dataset (Supervised)
Target video dataset (Unsupervised)
Generic detector (Input)
Domain adaptation algorithm (Proposed)
Scene-specific detector (Output)

(a) (b)

(c) (d)

Fig. 1. The goal of this paper. (a) shows the high level view of this paper. We have a generic detector trained on a generic dataset (c) available. The proposed algorithm automatically adapts it to different target scenes given only videos of those scenes without any label information attached with the videos. (b) & (d) show the improvement of the automatically adapted scene-specific detector over the generic detector for two target scenes. For (b) & (d), the left figure shows the detection results for the generic detector and the right for the adapted detector. For visualisation, all detection results shown are thresholded at around 1 False Positive Per Image (FPPI).

For videos, apart from the structural assumption about the data distributions, there is knowledge that can be exploited and is unique to videos such as the ability to model long-term scene background information to infer about foreground objects and the knowledge that objects in videos move in a smooth and spatially and temporally coherent manner (which, for example, allows object tracking). In fact, our proposed algorithm makes effective use of this rich spatio-temporal "scene" knowledge and we show that it can render the task of detector adaptation for videos much easier.

The rest of the paper is organized as follows. In Section II, we list the contributions of this paper. Section III reviews the related literature. In Section IV, we describe the overview and the details of the proposed algorithm. Section V discusses the experiments and the results. Finally, Section VI gives the conclusion.

II. Contribution

The contribution of this paper is four-fold. Firstly, we introduce the idea of *bounding box proposals* and *initial verification* for efficient generation of a large number of scene-specific pedestrian data with high probability of accuracy. Secondly, we use short-term tracking for spatio-temporal verification and data *expansion* (*i.e.* collection of hard pedestrian data). Thirdly, we show that this combination of bounding box proposal and initial verification, spatio-temporal verification and expansion does not need any iterative self-training rounds, effectively making it a *non-iterative* algorithm. Despite that, our algorithm can compete or even outperform state-of-the-art self-training algorithms. Fourthly, unlike most state-of-the-art algorithms, our algorithm does not require the generic dataset for detector adaptation: just the generic detector alone is sufficient. There are many advantages to this. For instance, the generic dataset may not be available for certain reasons (such as due to copyright restrictions). Or it may be costly to transmit the generic dataset to different sites for detector adaptations for many different scenes.

III. Related work

Although a considerable amount of research has been done in domain adaptation for audio and text [6], [5], domain adaptation for images and video (pedestrian detection in particular) is a relatively new area. Domain adaptation for object *detection* (as opposed to object *classification*) in videos brings about new challenges (such as having to deal with large amounts of data and class imbalance) and also opportunities that we can exploit (*e.g.* from videos, we can learn a lot of things from the scene to help with the domain adaptation). Much of the state-of-the-art research uses iterative self-training in one form or another [7], [8], [9], [10], [11]. In order to adapt a generic pedestrian detector to a specific scene, a typical system would run the generic detector on some frames in a video, then score each detection using some heuristics and then add the most confident positive and negative detections to the original dataset for retraining. This is typically repeated over many iterations. But this approach is prone to drifting since wrongly labelled examples in one iteration could make the detector become progressively worse in the following iterations.

Nair and Clark [11] propose an online classifier which automatically tunes itself to a restricted area of an *indoor* scene by iterative retraining using the results of background subtraction. Grabner *et al.* [12] propose to train one classifier for each image location and use a fixed rule to update the classifiers: positive examples are assumed known and fixed at the beginning and all incoming unlabelled data are treated as negative examples. But this may cause drifting if an object remains stationary for too long, causing false negative updates to accumulate. Roth *et al.* [4] extended this approach by building generative models for positive and negative classes, which are then combined to obtain a discriminative classifier. Stalder *et al.* [13] further extended this by using local pools for positive and negative samples for each grid location instead of a fixed positive set used in [4], [12]. But the goal of their method and this paper is different in that our approach does not require any manual ground-plane input or 3D context and we are interested in adapting a detector from a generic dataset to a target scene with the feature extraction and classification mechanism (and hence, the number of classifiers) held *fixed*. What we are looking for is the relative improvement in performance compared to the generic detector.

Wang *et al.* [14] extract dense features from detections of the generic detector to build a vocabulary tree, and examples with high confidence are sparsely coded to build the scene-specific detector. Their proposed approach requires manual setting of several sensitive thresholds and also makes use of the assumption that a detected object should be detected with high confidence at least once among all frames considered. However, this assumption is not practical in many situations. For unsupervised domain adaptation in images, Gopalan *et al.* [15] and Gong *et al.* [16] propose approaches to characterize the domain shift between the source and target datasets. There have also been approaches based on co-training [17] and reclassifying points near the decision boundary [18] by treating each new image as a new domain and using Gaussian process regression.

Wang and Wang [8] iteratively improve a generic pedestrian detector by selecting new confident examples to add to the original dataset for retraining at every iteration. In order to obtain confident examples, they use a combination of vehicle and pedestrian paths, multiple different cues such as bounding box locations and sizes, background subtraction, thresholds, filters and hierarchical clustering. Their approach requires some parameter tuning such as deciding the length of a video segment and hyper-parameters when learning the topic model, etc. There is also a need to manually label the discovered paths and an assumption that pedestrians and vehicle paths are not overlapped to a certain degree. They extended their method in [7] by incorporating techniques such as reweighting the source data, confidence propagation and using the confidence when retraining rather than hard thresholding.

IV. Our approach

A. Overview

The overview of our algorithm is illustrated in Fig. 2. We describe the algorithm briefly below and the details of the algorithm are explained in the following sections. The inputs to the algorithm are a generic detector C_g and a target video $\mathbf{V} = [I_1, I_2, \ldots, I_N]$ of N frames to adapt C_g to. The desired output is a scene-specific detector C_s. The first step involves bottom-up generation of *bounding box proposals* of pedestrians for $\mathbf{V}$. Then these bounding box proposals are verified using C_g (*initial verification*). The result is a set of *verified proposals*. Each of these verified proposals is tracked for a short period (*e.g.* for 3 seconds). Then each track is verified using C_g and majority voting (*spatio-temporal verification*). For each verified track, the first data in the track and all the data which give negative labels (*i.e.* hard positives) are collected. All these *expanded* data are pooled across all the verified tracks to form the positive data for the scene-specific detector. Negative data for the scene-specific detector are sampled from the regions in $\mathbf{V}$ which C_g classifies as negative *and* which do not overlap with any areas of the bounding box proposals. After the scene-specific positive and negative data have been obtained, the scene-specific detector C_s is trained. The approach is formalized in Algorithm 1.

Algorithm 1 Non-iterative detector adaptation

```
Input: ${C_g, \mathbf{V}}$
Output: $C_s$

% Generate bounding box proposals %

Bounding box proposals, $\mathcal{B} \leftarrow \varnothing$
for $I_i \in \mathbf{V}$ do
    Let $I_{\text{model}}$ be the previous estimate of scene background
    $I_{\text{model}} \leftarrow \text{UpdateBGModel}(I_i, I_{\text{model}})$
    $I_{\text{fgmask}} \leftarrow$ background subtraction on $\{I_i, I_{\text{model}}\}$
    Perform connected component analysis on $I_{\text{fgmask}}$
    $\mathcal{B} \leftarrow \mathcal{B} \cup$ bounding boxes for connected blobs
end for

% Initial verification %

Verified proposals, $\mathcal{B}_v \leftarrow \varnothing$
Let $F$ be the function for resizing and feature extraction
for $b_i \in \mathcal{B}$ do
    score $= C_g(F(b_i))$
    if score $> 0$ then
        $\mathcal{B}_v \leftarrow \mathcal{B}_v \cup b_i$
    end if
end for

% Spatio-temporal verification & expansion %

Scene-specific positive data, $\mathcal{D}_p \leftarrow \varnothing$
for $v_i \in \mathcal{B}_v$ do
    Set of tracked patches, $P \leftarrow \text{ShortTermTrack}(v_i)$
    Classifier scores, $S \leftarrow \varnothing$
    for $p_i \in P$ do
        $S \leftarrow S \cup C_g(F(p_i))$
    end for
    if $\text{MajorityVote}(S) = \text{positive}$ then
        for $p_i \in P$ do
            if $i = 1$ or $C_g(F(p_i)) \leq 0$ then
                $\mathcal{D}_p \leftarrow \mathcal{D}_p \cup F(p_i)$
            end if
        end for
    end if
end for

% Collect scene-specific negative data %

$\mathcal{D}_n \leftarrow \varnothing$
for $I_i \in \mathbf{V}$ do
    Let $W$ be the set of sliding window patches on $I_i$
    $W \leftarrow W \cup \{w \in W : (C_g(F(w)) > 0) \wedge (w \cap \mathcal{B} = \varnothing)\}$
    $\mathcal{D}_n \leftarrow \mathcal{D}_n \cup F(W)$
end for

% Train scene-specific detector %

$C_s \leftarrow$ Train classifier on $\{\mathcal{D}_p, \mathcal{D}_n\}$

return $C_s$
```

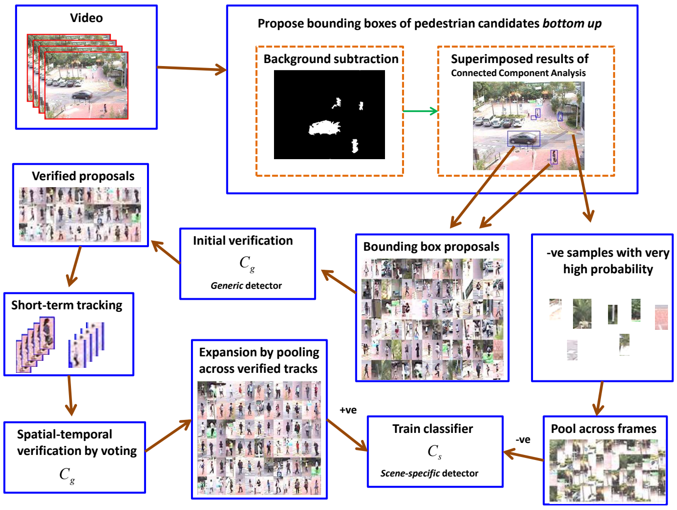

Video
Propose bounding boxes of pedestrian candidates *bottom up*
Background subtraction
Superimposed results of
Connected Component Analysis
Verified proposals
Initial verification
C_g
Generic detector
Bounding box proposals
-ve samples with very high probability
Short-term tracking
Expansion by pooling across verified tracks
+ve
Spatial-temporal verification by voting
C_g
Train classifier
C_s
Scene-specific detector
-ve
Pool across frames

Fig. 2. Overview of our proposed approach.

B. Generating bounding box proposals

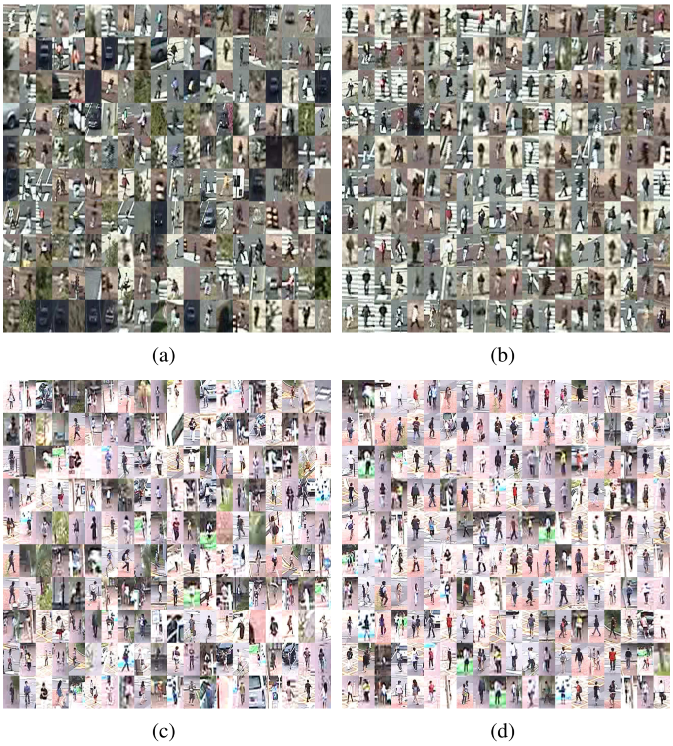
The first step of the algorithm is to propose bounding boxes of pedestrian candidates in a bottom up fashion. We do this by performing a *background subtraction* algorithm on $\mathbf{V}$ and *connected component analysis* on the resulting foreground pixels and get tight bounding boxes around the connected components. These *bounding box proposals* across all the frames are *pooled* and stored in the set $\mathcal{B}$.

C. Initial verification

We go through each bounding box proposal $b_i \in \mathcal{B}$, get the image patch underlying it, resize the patch and extract the features using the given feature extraction function F. These features are then passed to the generic detector for scoring. If the score is positive (*i.e.* the prediction is a pedestrian), b_i is considered a *verified proposal* and is stored in $\mathcal{B}_v$. The combination of the background proposal and initial verification stages efficiently samples a high number of pedestrian patches with high probability. Random samples from sets $\mathcal{B}$ and $\mathcal{B}_v$ are shown in Fig. 3. We can observe that $\mathcal{B}_v$ barely contains any mistakes. This is because the error introduced by bounding box proposal and initial verification are *uncorrelated*. In addition, $\mathcal{B}_v$ consists of very accurately localised pedestrians (i.e. no patch alignment errors, etc.) which are suitable for training a detector.

D. Spatio-temporal verification & expansion

Although the set $\mathcal{B}_v$ is large and contains pedestrians with high probability, one might argue that $\mathcal{B}_v$ may be biased towards pedestrians that the generic detector is already "good" at and it might also contain some errors which both the proposal generation and initial verification stages could not eliminate. Therefore, we spatio-temporally verify and *expand* the set $\mathcal{B}_v$ by short-term tracking each verified proposal $v_i \in \mathcal{B}_v$ producing a *tracklet*. The short-term tracking is done independently for each v_i. We do not use the generic detector *during* tracking, instead we use an online learnt appearance model. *Not* using the generic detector during tracking allows us to *decouple* the errors made by the generic detector and


(a)

(b)

(c)

(d)

Fig. 3. 1st and 2nd rows correspond to MIT traffic and CUHK Square datasets respectively. 1st column shows 200 random samples from set $\mathcal{B}$ and 2nd column shows 200 random samples from set $\mathcal{B}_v$ (see Section IV for notation).

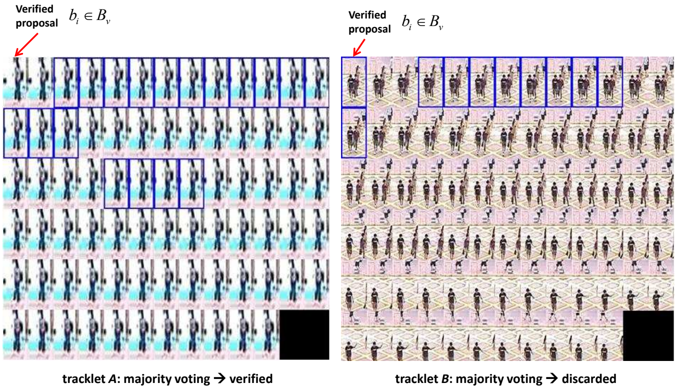

Verified proposal
$b_i \in B_v$
Verified proposal
$b_i \in B_v$
tracklet A: majority voting → verified
tracklet B: majority voting → discarded

Fig. 4. Visualization of two example tracklets obtained from tracking two verified proposals respectively. In each tracklet P, we show the patches p_i belonging to P. Since each tracklet is about 3-seconds long (76-frames), there are 76 patches in a tracklet (with the first patch being the verified proposal). For tracklet A, the blue rectangles indicate patches which the generic detector C_g classifies as non-pedestrians, *i.e.* they are false negatives. However, spatio-temporal verification (by majority voting in the track) successfully verifies the track as a pedestrian track. The patches with blue rectangles are therefore "hard" positives and are collected, along with the verified proposal, as scene-specific positive data. The patches with the blue rectangles can be termed as the *expansion* of the verified proposal. On the other hand, spatio-temporal verification on tracklet B correctly discards the track. Even though the first patch (the verified proposal) and other patches with the blue rectangles are erroneously classified by C_g as pedestrians (when in fact, each has 2 pedestrians in it), majority voting successfully discards the entire track along with the verified proposal.

the tracker. Spatio-temporal verification is done by applying the generic detector on each patch in the track and taking the majority vote of the classification scores. If the majority vote is positive, then we consider the tracklet as *verified* and add the data in the tracklet to the collection of scene-specific positive data. In order to avoid the number of data from getting too large, instead of adding every patch in a verified tracklet as new positive data, we add only the hard positives, *i.e.* patches in the track which have negative classification scores. Fig. 4 illustrates the idea.

E. Collecting negative examples

Scene-specific negative data are randomly sampled from all possible multi-scale sliding window bounding box regions that neither overlap with the bounding box proposals nor with pedestrian detections in the frames of the video. This way, negative data with high confidence can be obtained.

F. Training the scene-specific detector

Now that we have collected positive and negative scene-specific data ($\mathcal{D}_p$ and $\mathcal{D}_n$ respectively), we obtain a scene-specific detector C_s by training a classifier on $\mathcal{D}_p$ and $\mathcal{D}_n$. It should be noted that $\mathcal{D}_p$ and $\mathcal{D}_n$ are entirely made up of data from the target scene only and by doing this, we are effectively setting the weights for the source dataset to zero. We show the effectiveness of throwing away the source dataset in Section V.

G. Extra analysis on bounding box proposal and verification

As discussed previously, the combination of bounding box proposal generation and initial verification are the first and second steps of our algorithm respectively. We now focus on this combination and compare it to state-of-the-art research [11], [8], [7] which use background subtraction as (one of the steps in) verification of detections of the generic detector C_g. This is different from our approach because we do *not* use background subtraction as a *verifier*: instead it is the other way round in our algorithm: C_g is *the* verifier of background subtraction proposals. Although this is a simple modification, it makes a significant difference in performance as shown in Section V. Essentially, we are not using C_g as a sliding window "detector", but we are making its task easier by using it as a classifier on the verified proposals. This minimizes possible errors introduced by a sliding window detector (such as sliding window space discretization error and non-maxima suppression error). Furthermore, if we use a real-time state-of-the-art background subtraction algorithm for bounding box proposal generation, we can very quickly and efficiently obtain a large number of verified proposals. It should be also noted that this is *not* the same as the approach taken in hierarchical segmentation-based selective search strategy to speed up object detection (*e.g.* [19]). The difference is that their goal is improving the detector per-se whereas our goal is domain adaptation and *not* to detect every single pedestrian (i.e. high recall with high precision) *during* the adaptation stage (instead, part of our goal during the adaptation stage is just to collect a reasonable amount of confident pedestrians). Furthermore, they are working on static images and hierarchical segmentation based on color and texture cues whereas in our algorithm, we are directly using cues from the video without any hierarchical segmentation. Lastly, we do not apply any background subtraction or any other kinds of segmentation during the test stage after the detector adaptation.

V. Experimental Results

A. *Classifier & features*

For feature extraction and classification, we use Histogram of Oriented Gradients (HOGs) [20] and linear SVM respectively. This is done for simplicity and our algorithm can in principal be used with other feature extraction techniques and classifiers.

B. *Datasets*

We use the INRIA pedestrian dataset [20] as the generic dataset. We use two challenging public video datasets as target scenes: a 90-minutes long MIT Traffic dataset [8] and a 60-minutes long CUHK Square dataset [7]. We use similar settings and ground truth as given in those datasets for evaluation purposes only. For each video (dataset), we divide it into two roughly equal parts:

1) 1st half (*adaptation stage*): Used for (unsupervised) training. This is where all the detector adaptation takes place. No manual annotation is used for any detector adaptation.
2) 2nd half (*testing stage*): After the detector adaptation is performed in the first half, the second half is used for testing. 100 frames are uniformly sampled and groundtruth is annotated for evaluation. In this stage, no background subtraction, ground-plane assumption or other cues are used. Only pure (sliding window) detection performance is evaluated. The detector is applied independently on each frame being evaluated.

C. *Descriptions of experiments*

Evaluation is performed in terms of recall-FPPI (False Positives Per Image) curves. The PASCAL 50% overlap criteria [21] is used to score the detection bounding boxes. Six different types of experiments are performed:

1) `Proposed`: Our proposed algorithm.
2) `Baseline(Generic)`: The detector trained on the generic dataset. This is the baseline for our comparison.
3) `Nair (CVPR 04)`: This is an iterative self-training algorithm for detector adaptation using background subtraction [11].
4) `Wang (CVPR 11)`: The detector adaptation algorithm that uses multiple cues in the video [8].
5) `Wang (CVPR 12)`: Another state-of-the-art algorithm presented in [7] and is an extension of [8].
6) `Human supervision(`X`)`: Fully-supervised scene-specific detector obtained by manually annotation on X number of uniformly sampled frames in the 1st half of video. Different values of X are experimented.
7) `Proposed + source dataset (INRIA)`: A modification of our proposed algorithm (`Proposed`). Instead of training the scene-specific detector only on the collected scene-specific data, we also include the generic dataset for training.

D. *Evaluation*

Performance curves are shown in Fig. 5 with their plot legends referring to the types of experiments described previously. We discuss them below.

1) Comparison with generic detector: We see that our proposed method `Proposed` has a much higher performance than the generic detector `Baseline(Generic)` in all the experiments in both datasets. For CUHK Square, at 1 FPPI, the recall of `Proposed` is about three times that of `Baseline(Generic)`, which is a significant improvement. For MIT traffic, the recall of `Proposed` is about 3.5 times higher than the recall of `Baseline(Generic)` at 1 FPPI. This shows that it is worthwhile to run the detector adaptation algorithm whenever we have a new scene and we want to automatically generate a much better detector than the generic detector.

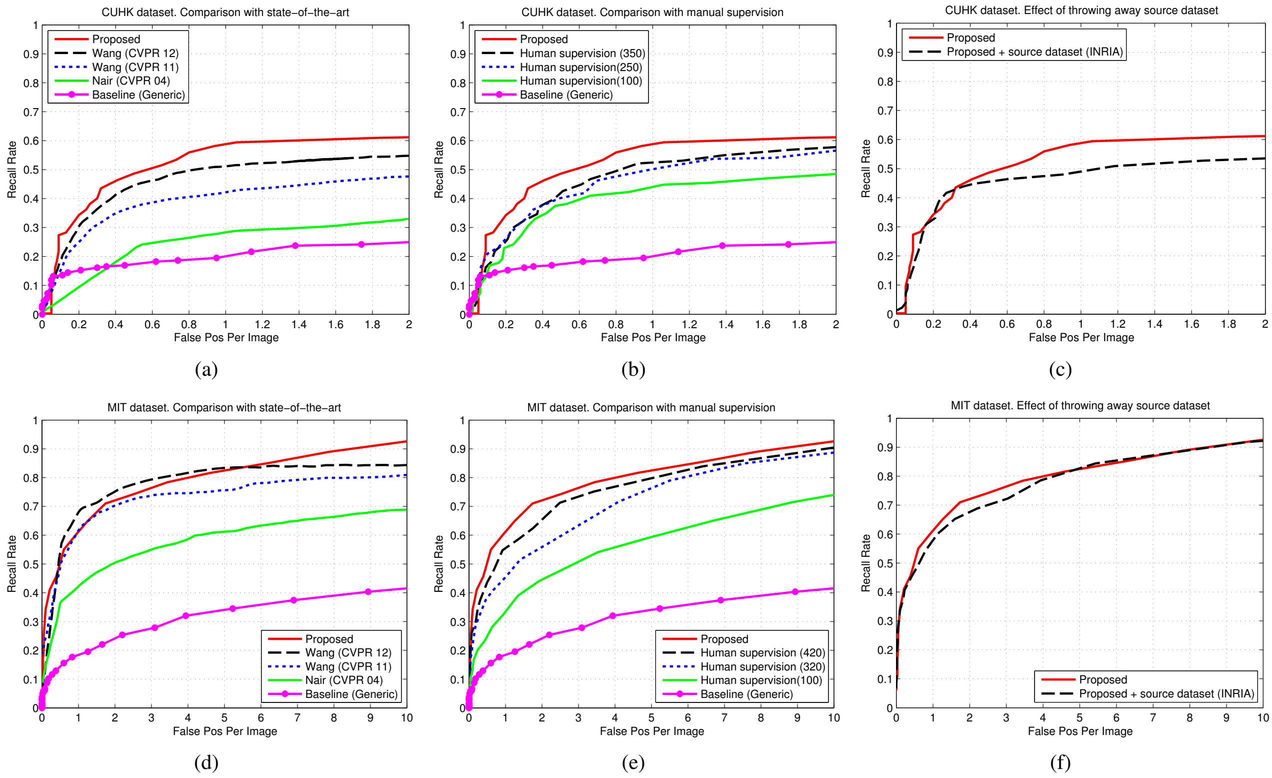
2) Comparison with state-of-the-art: This is shown is Fig. 5 (a) & (d) for CUHK and MIT datasets respectively. For CUHK, our non-iterative algorithm `Proposed` clearly outperforms all the state-of-the-art self-training approaches, `Wang(CVPR12)`, `Wang(CVPR11)` and `Nair(CVPR04)`, which require a manually set number of iterations to reach their peak performance given in the graphs. For MIT, `Proposed` competes well with `Wang(CVPR12)` and is better than both `Wang(CVPR11)` and `Nair(CVPR04)`. In both datasets, `Proposed` is significantly better than `Nair(CVPR04)` showing that our algorithm, despite using background subtraction in a major way, has a much higher performance due to it being a novel combination of bounding box proposal, initial verification, spatio-temporal verification and expansion by tracking.

3) Comparison with human supervision: The performance curves for detectors trained with various amounts of human supervision is shown in Fig. 5 (b) & (e). For CUHK, `Proposed` outperforms all the detectors trained with manual human supervision including the one that was trained with 350 frames worth of manual annotation. For MIT, similar observations can be made. However, as the number of frames which are manually annotated increases to a sufficient number, it is expected that `Human supervision(`X`)` may reach or go higher than the performance of `Proposed`.

4) Effect of throwing away the source dataset: Our algorithm does not require the source dataset when training the scene-specific detector. The effect of including the source data is shown in Fig. 5 (c) & (f). It is observed that for both datasets, incorporating the source dataset slightly decreases the performance. This observation is consistent with the intuition that adding source dataset is akin to trying to do well on *both* the generic dataset and target scene, with the net result that the scene-specific is less well tuned to the target scene.

VI. Conclusion

In this paper, we propose an efficient and automatic non-iterative algorithm that adapts a generic detector to a specific scene given only the unlabelled video of the scene. The algorithm outputs a scene-specific detector that performs much better than the generic detector and performs as well as fully supervised detectors trained on hundreds of frames of manual annotations. Moreover, experimental results show that the algorithm outperforms state-of-the-art approaches on two challenging datasets. The scene-specific detector generated by our algorithm could be used as a building block for high level scene understanding and to improve tracking-by-detection applications.



(a) (b) (c)

(d) (e) (f)

Fig. 5. Detection performance curves (all on testing datasets). 1st row shows results for CUHK Square dataset. 2nd row shows results for MIT Traffic dataset. 1st column gives comparison of our proposed algorithm with state-of-the-art approaches. The 2nd column compares with manual annotation and the 3rd column shows the effect of throwing away the source dataset.

REFERENCES

[1] P. Dollár, C. Wojek, B. Schiele, and P. Perona, “Pedestrian detection: An evaluation of the state of the art,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 4, pp. 743–761, 2012.

[2] M. Enzweiler and D. M. Gavrila, “Monocular pedestrian detection: Survey and experiments,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 12, pp. 2179–2195, 2009.

[3] P. Dollár, C. Wojek, B. Schiele, and P. Perona, “Pedestrian detection: A benchmark,” in *CVPR*, 2009, pp. 304–311.

[4] P. M. Roth, S. Sternig, H. Grabner, and H. Bischof, “Classifier grids for robust adaptive object detection,” in *CVPR*, 2009, pp. 2727–2734.

[5] J. Jiang, “A literature survey on domain adaptation of statistical classifiers,” *Rapport interne, Comp. Sc. Dep. at Univ. of Illinois. sifaka. cs. uiuc. edu/jiang4/domain_adaptation/survey/da_survey. pdf*, 2008.

[6] H. D. III, “Frustratingly easy domain adaptation,” *CoRR*, vol. abs/0907.1815, 2009.

[7] M. Wang, W. Li, and X. Wang, “Transferring a generic pedestrian detector towards specific scenes,” in *CVPR*, 2012, pp. 3274–3281.

[8] M. Wang and X. Wang, “Automatic adaptation of a generic pedestrian detector to a specific traffic scene,” in *CVPR*, 2011, pp. 3401–3408.

[9] C. Rosenberg, M. Hebert, and H. Schneiderman, “Semi-supervised self-training of object detection models,” in *WACV/MOTION*, 2005, pp. 29–36.

[10] L. F.-F. D. K. Kevin Tang, Vignesh Ramanathan, “Shifting weights: Adapting object detectors from image to video,” in *Neural Information Processing Systems (NIPS)*, 2012.

[11] V. Nair and J. J. Clark, “An unsupervised, online learning framework for moving object detection,” in *CVPR (2)*, 2004, pp. 317–324.

[12] H. Grabner, P. M. Roth, and H. Bischof, “Is pedestrian detection really a hard task?” in *Proc. IEEE Intern. Workshop on Performance Evaluation of Tracking and Surveillance*, 2007, pp. 1–8.

[13] S. Stalder, H. Grabner, and L. Gool, “Exploring context to learn scene specific object detectors,” in *Proc. PETS*, 2009.

[14] X. Wang, G. Hua, and T. X. Han, “Detection by detections: Non-parametric detector adaptation for a video,” in *CVPR*, 2012, pp. 350–357.

[15] R. Gopalan, R. Li, and R. Chellappa, “Domain adaptation for object recognition: An unsupervised approach,” in *ICCV*, 2011, pp. 999–1006.

[16] B. Gong, Y. Shi, F. Sha, and K. Grauman, “Geodesic flow kernel for unsupervised domain adaptation,” in *CVPR*, 2012, pp. 2066–2073.

[17] O. Javed, S. Ali, and M. Shah, “Online detection and classification of moving objects using progressively improving detectors,” in *CVPR (1)*, 2005, pp. 696–701.

[18] V. Jain and E. G. Learned-Miller, “Online domain adaptation of a pre-trained cascade of classifiers,” in *CVPR*, 2011, pp. 577–584.

[19] K. E. A. Van de Sande, J. R. R. Uijlings, T. Gevers, and A. Smeulders, “Segmentation as selective search for object recognition,” in *Computer Vision (ICCV), 2011 IEEE International Conference on*, 2011, pp. 1879–1886.

[20] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *CVPR (1)*, 2005, pp. 886–893.

[21] M. Everingham, L. J. V. Gool, C. K. I. Williams, J. M. Winn, and A. Zisserman, “The pascal visual object classes (voc) challenge,” *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.