Automatic adaptation of object detectors to new domains using self-training

Aruni RoyChowdhury Prithvijit Chakrabarty Ashish Singh SouYoung Jin
Huaizu Jiang Liangliang Cao Erik Learned-Miller
College of Information and Computer Sciences
University of Massachusetts Amherst
{arunirc, pchakrabarty, ashishsingh, souyoungjin, hzjiang, llcao, elm}@cs.umass.edu

Abstract

This work addresses the unsupervised adaptation of an existing object detector to a new target domain. We assume that a large number of unlabeled videos from this domain are readily available. We automatically obtain labels on the target data by using high-confidence detections from the existing detector, augmented with hard (misclassified) examples acquired by exploiting temporal cues using a tracker. These automatically-obtained labels are then used for re-training the original model. A modified knowledge distillation loss is proposed, and we investigate several ways of assigning soft-labels to the training examples from the target domain. Our approach is empirically evaluated on challenging face and pedestrian detection tasks: a face detector trained on WIDER-Face, which consists of high-quality images crawled from the web, is adapted to a large-scale surveillance data set; a pedestrian detector trained on clear, daytime images from the BDD-100K driving data set is adapted to all other scenarios such as rainy, foggy, night-time. Our results demonstrate the usefulness of incorporating hard examples obtained from tracking, the advantage of using soft-labels via distillation loss versus hard-labels, and show promising performance as a simple method for unsupervised domain adaptation of object detectors, with minimal dependence on hyper-parameters[1].

1. Introduction

The success of deep neural networks has resulted in state-of-the-art object detectors that obtain high accuracy on standard vision benchmarks (*e.g*. MS-COCO [36], PASCAL VOC [12], etc.), and are readily available for download as out-of-the-box detection models [17, 23]. However, it is unrealistic to expect a single detector to generalize to *every* domain. Due to the data-hungry nature of supervised training of deep networks, it would require a lot of labeling efforts to re-train a detector in a completely supervised manner for a new scenario.

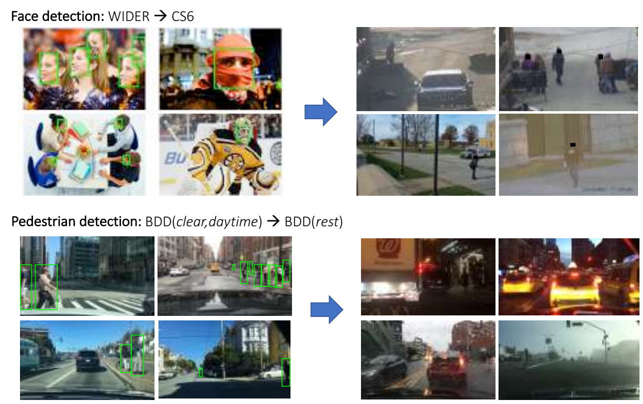


Figure 1: **Unsupervised cross-domain object detection.** *Top:* adapting a face detector trained on labeled high-quality web images from WIDER-Face [67] to unlabeled CS6/IJB-S [29] video frames. *Bottom:* adapting a pedestrian detector trained on labeled images from the *(clear, daytime)* split of the BDD-100k dataset [68] to unlabeled videos from all the other conditions (*e.g*. night-time, foggy, rainy, *etc*.).

This paper considers the following problem: Given an off-the-shelf detector, can we let it automatically improve itself by watching a video camera? We hope to find a new algorithm based on unsupervised self-training that leverages large amounts of readily-available unlabeled video data, so that it can relieve the requirement of labeling effort for the new domain, which is tedious, expensive, and difficult to scale up. Such a solution may be very useful to generalize existing models to new domains without supervision, *e.g*. a pedestrian detection system trained on imagery of US streets can adapt to cities in Europe or Asia, or help an off-the-shelf face detector improve its performance on video footage. Such an algorithm would be a label efficient solution for large-scale domain adaptation, obviating the need for costly bounding-box annotations when faced with a new

[1]Code and models are available at http://vis-www.cs.umass.edu/self-train/

domain.

Recent approaches to unsupervised domain adaptation in deep networks have attempted to learn domain invariant features through an adversarial domain discriminator [8, 15, 16, 60, 22], or by transforming labeled source images to resemble the target domain using a generative adversarial network (GAN) [24, 69, 5]. *Self-training* is a comparatively simpler alternate strategy, where the off-the-shelf model's predictions on the new domain are regarded as "pseudo-labeled" training samples [32, 7, 4, 65]; however this approach would involve re-training using significantly noisy labels. It becomes even more challenging when we consider object detectors in particular, as the model may consider a wrongly-labeled instance as a hard example [51] during training, and expend a lot of efforts trying to learn it.

In this paper, we leverage two types of information that is useful for object detection. First, object detectors can benefit from learning the temporal consistency in videos. Some hard cases missed by the detector could be recognized if the object is detected in neighboring frames. We combine both tracking and detection into one framework, and automatically refine the labels based on detection and tracking results. Second, there are examples of varying difficulty in the new domain, and we propose a distillation-based loss function to accommodate this relative ordering in a flexible fashion. We design several schemes to assign soft-labels to the target domain samples, with minimal dependence on hyper-parameters. We evaluate our methods for improving *single-image* detection performance without labels on challenging face and pedestrian detection tasks, where the target domain contains a large number of unlabeled videos. Our results show that training with soft labels improves over the usual hard (*i.e*. 0 or 1) labels, and reaches comparable to better performance relative to adversarial methods without extra parameters.

2. Related Work

Semi-supervised learning. Label-efficient semi-supervised methods of training object recognition models have a long history in computer vision [46, 63, 2, 49, 33, 14]. For a survey and empirical comparison of various semi-supervised learning methods applied to deep learning, we refer the reader to Odena *et al*. [42]. We focus on the *self-training* approach [7, 4, 65, 32], which involves creating an initial baseline model on fully labeled data and then using this model to estimate labels on a novel weakly-labeled or unlabeled dataset. A subset of these estimated labels that are most likely to be correct are selected and used to re-train the baseline model, and the process continues in an incremental fashion [41, 34, 39, 25, 66]. In the context of object detection, Rosenberg *et al*. [46] used the detections from a pre-trained object detector on unlabeled data as pseudo-labels and then trained on a subset of this noisy labeled data in an incremental re-training procedure. Recently, the *data distillation* approach [43] aimed to improve the performance of fully-supervised state-of-the-art detectors by augmenting the training set with massive amounts of pseudo-labeled data. In their case, the unlabeled data was from the same domain as the labeled data, and pseudo-labeling was done by selecting the predictions from the baseline model using test-time data augmentation. Jin *et al*. [26] use tracking in videos to gather *hard examples* – *i.e*. objects that fail to be detected by an object detector (false negatives); they re-train using this extra data to improve detection on still images. Our work shares the latter's strategy of exploiting temporal relationships to automatically obtain hard examples, but our goal is fundamentally different – we seek to *adapt* to a new target domain, while Jin *et al*. use the target domain to mine extra training samples to improve performance back in the source domain. We note that improvements in network architecture specific to video object recognition [13, 62] are orthogonal to our current motivation. Closely related work from Tang *et al*. [38] adapt a multi-person tracker model to test sequences by re-training on automatically-extracted high-quality samples.

Hard examples. Emphasizing difficult training samples has been shown to be useful in several works – *e.g*. online hard example mining (OHEM) [51], boosting [48]. Weinshall and Amir [64] show that for certain problem classes, when we do not have access to an optimal hypothesis (*e.g*. a teacher), training on examples the current model finds difficult is more effective than a self-paced approach which trains first on easier samples.

Unsupervised domain adaptation. There has been extensive work in addressing the shift between source and target domains [19, 3, 53] (see Csurka [9] for a recent survey). Some approaches try to minimize the Maximum Mean Discrepancy [19, 61, 37] or the CORAL metric [54] between the distribution of features from the two domains. Another popular direction is an adversarial setup, explored by recent works such as ADDA [60], CyCADA [22], gradient reversal layer (ReverseGrad) [16, 15], wherein the discriminator tries to predict the domain from which a training sample is drawn, and the model attains domain invariance by trying to fool this discriminator, while also learning from labeled source samples. In particular, the work of Tzeng *et al*. [59] obtains soft-labels from model posteriors on *source domain* images, aiming to transfer inter-category correlations information across domains. Our soft-labels, on the other hand, are obtained on the *target domain*, have only a single category (therefore inter-class information is not applicable), and aims at preserving information on the relative difficulty of training examples across domains.

Cross-domain object detection. The domain shift [30] of detectors trained on still images and applied to video frames has been addressed in several works, mostly relying on some form of weak supervision on the target domain and selecting target samples based on the baseline detector confidence score [20, 57, 50, 11, 31, 6]. Several approaches have used weakly-labeled video data for re-training object detectors [28, 52, 57]. Our work is motivated in particular by Tang *et al*. [57], who use tracking information to get pseudo-labels on weakly-labeled video frames and adopt a curriculum-based approach, introducing easy examples (*i.e*. having low loss) from the target video domain into the re-training of the baseline detector. Despite the common motivation, our work differs on two major points – (a) we show the usefulness of combining *both* hard and easy examples from the target domain when re-training the baseline model, and (b) using the knowledge distillation loss to counter the effect of label noise. Jamal *et al*. [1] address the domain shift between various face detection datasets by re-calibrating the final classification layer of face detectors using a residual-style layer in a low-shot learning setting. Two recent methods [24, 8] for *domain-adaptive object detection* are particularly relevant to our problem. The weakly-supervised method of Inoue *et al*. [24] first transforms the labeled source (natural) images to resemble the target images (watercolors) using the CycleGAN [69], fine-tunes the baseline (pre-trained) detector on this "transformed source" data, and then obtains pseudo-labels on the target domain using this domain-adapted model. The task of image generation is fairly difficult, and we posit that it may be possible to address domain adaptation without requiring a generative model as an intermediate step. The fully unsupervised method of Chen *et al*. [8] learns a domain-invariant representation by using an adversarial loss from a *domain discriminator* [15, 16] at various levels of the Faster R-CNN architecture, showing significant improvements when adapting to challenging domain shifts such as clear to foggy city scenes, simulated to real driving videos, *etc*. While a powerful approach, the design of new discriminator layers and adversarial training are both challenging in practice, especially without a labeled validation set on the target domain (as is the case in an unsupervised setting). Similarly motivated approaches have been applied to adapting to foggy street scenes in a progressive manner [47, 10].

3. Proposed Approach

Automatically labeling the target domain is described in Sec. 3.1, re-training using these pseudo-labels in Sec. 3.2 and creating soft-labels in Sec. 3.3.

3.1. Automatic Labeling of the Target Domain

Self-labeling [58] or pseudo-labeling [32] adapts a pre-existing or *baseline* model, trained on a labeled *source* domain $\mathcal{S}$, to a novel unlabeled *target* domain $\mathcal{T}$, by treating the model's own predictions on the new dataset as training labels. In our case, we obtain target domain pseudo-labels by selecting high-confidence predictions of the baseline detector, followed by a refinement step using a tracker.

Pseudo-labels from detections. The baseline detector is run on every frame of the unlabeled videos in the target domain and if the (normalized) detector confidence score for the i-th prediction (*i.e*. the model's posterior), d_i, is higher than some threshold θ, then this prediction is added to the set of pseudo-labels. In practice, we select 0.5 for θ for face detection and 0.8 for person detection. Note that such a threshold is easily selected by visually inspecting a small number of unlabeled videos from $\mathcal{T}$ (5 videos); we compare with a fully-automated procedure in Sec. 4.6.

Refined labels from tracking. Exploiting the temporal continuity between frames in a video, we can enlarge our set of pseudo-labels with objects missed by the baseline detector. To link multiple object detections across video frames into temporally consistent tracklets, we use the algorithm from Jin *et al*. (Sec. 3 of [27]) with the MD-Net tracker [40]. Now, given a tracklet that consistently follows an object through a video sequence, when the object detector did not fire (*i.e*. $d_i < \theta$) in some difficult frames, the tracker can still correctly predict an object (see Fig. 2(a)). We expand the set of pseudo-labels to include these "tracker-only" bounding-boxes that were missed by the baseline detector, since these *hard examples* are expected to have a larger influence on the model's decision boundary upon retraining [55, 51, 26]. Further, we prune out extremely short tracklets (less than 10 frames) to remove the effects caused by spurious detections.

3.2. Training on pseudo-labels

We use the popular Faster R-CNN (FRCNN) [45, 44] as our detector. In a naive setting, we would treat both labeled source-domain data and pseudo-labeled target-domain data identically in terms of the loss. We give a label of 1 to *all* the target domain pseudo-labeled samples, irrespective of whether it originated from the baseline detector or the tracker – *i.e*. for X_i, the i-th training sample drawn from $\mathcal{T}$, the label y_i is defined as

$$y_i = \begin{cases} 1, & \text{if } X_i \text{ is a } \textit{pos.} \text{ sample (from detector or tracker).} \\ 0, & \text{if } X_i \text{ is a } \textit{neg.} \text{ sample.} \end{cases} \quad (1)$$

Note that here X_i is not an image, but a *region* in an image. For training the classification branch, we use a binary cross-entropy loss on the i-th training sample:

$$\mathcal{L}_i(y_i, p_i) = -[y_i \log(p_i) + (1 - y_i)\log(1 - p_i)] \quad (2)$$

where "*hard*" label $y_i \in \{0, 1\}$ and the model's predicted posterior $p_i \in [0, 1]$. This is similar to the method of Jin *et*

[2] Some faces hidden following permissions in [29].

(a) (b)

Figure 2: **(a) Pseudo-labels from detection and tracking:**[2] In three consecutive video frames, high-confidence predictions from the baseline detector are marked in *green*, and faces missed by the detector (*i.e.* low detector confidence score) but picked up by the tracker are marked in *yellow*. **(b) Soft-labeling example:** baseline detector confidences are $d_1 = 0.78$, $d_2 = 0.83$, $d_3 = 0.32$; confidence threshold $\theta = 0.5$. Following Eqn 3, high-confidence detections (*green*) are assigned soft-scores $s_i = d_i$, *i.e.* $s_1 = 0.78$ and $s_2 = 0.83$. The tracker-only sample (*yellow*) has detector score below the threshold: $d_3 = 0.32 < \theta$. It gets soft-score $s_3 = \theta = 0.5$.

al. [26], which assigns a label of 1 for both easy and hard positive examples during re-training.

3.3. Distillation loss with soft labels

For training data coming from $\mathcal{T}$, many of the y_is can be noisy, so a "*soft*" version of the earlier $\{0, 1\}$ labels could help mitigate the risk from mislabeled target data. Label smoothing in this fashion has been shown to be useful in generalization [56, 21], in reducing the negative impact of incorrect training labels [35] and is more informative about the distribution of labels than one-hot encodings [59]. In our case, each target-domain *positive* label can have two possible origins – (i) high-confidence predictions from the baseline detector or (ii) the tracklet-formation process. We assign a ***soft score*** s_i to each positive target-domain sample $X_i \in \mathcal{T}$ as follows:

$$s_i = \begin{cases} d_i, & \text{if } X_i \text{ originates from detector.} \\ \theta, & \text{if } X_i \text{ originates from tracker.} \end{cases} \quad (3)$$

For a pseudo-label originating from the baseline detector, a high detector confidence score d_i is a reasonable measure of reliability. Tracker-only pseudo-labels, which could be objects missed by the baseline model, are emphasized during training – their soft score is raised up to the threshold θ, although the baseline's confidence on them had fallen below this threshold. An illustrative example is shown in Fig. 2(b).

Label interpolation. A ***soft label*** $\tilde{y}_i$ is formed by a linear interpolation between the earlier hard labels y_i and soft scores s_i, with $\lambda \in [0, 1]$ as a tunable hyper-parameter.

$$\tilde{y}_i = \lambda s_i + (1 - \lambda) y_i \quad (4)$$

The loss for the i-th positive sample now looks like

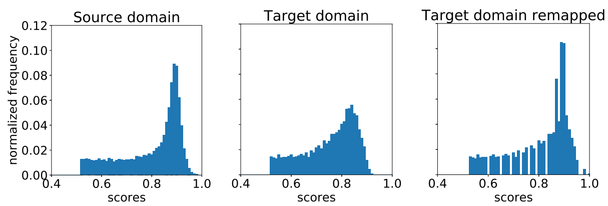
$$\mathcal{L}_i^{distill} = \begin{cases} \mathcal{L}_i(y_i, p_i), & \text{if } X_i \in \mathcal{S}. \\ \mathcal{L}_i(\tilde{y}_i, p_i), & \text{if } X_i \in \mathcal{T}. \end{cases} \quad (5)$$

Setting a high value of λ creates softer labels $\tilde{y}_i$, trusting the baseline source model's prediction s_i more than than the riskier target pseudo-labels y_i. In this conservative setting, the softer labels will decrease the overall training signal from target data, but also reduces the chance of incorrect pseudo-labels having a large detrimental effect on the model parameters.

We now describe two schemes to avoid explicitly depending on the λ hyper-parameter –

I. Constrained hard examples. Assigning a label of 1 to both "easy" and "hard" examples (*i.e.* high-confidence detections and tracker-only samples), as in Sec. 3.2, gives equal importance to both. Training with *just* the hard examples can be sub-optimal – it might decrease the model's posteriors on instances it was getting correct initially. Ideally, we would like to emphasize the hard examples, while simultaneously *constraining* the model to maintain its posteriors on the other (easy) samples. We can achieve this by setting $\theta = 1$ in Eq. 3 and $\lambda = 1$ in Eq. 4, which would create a label of 1 for tracker-only "hard" examples, and a label equal to baseline detector score for the high-confidence detections, *i.e.* "easy" examples.

II. Cross-domain score mapping. Let us hypothetically consider what the distribution of detection scores on $\mathcal{T}$ would be like, had the model been trained on *labeled* target domain data. With minimal information on $\mathcal{T}$, it is reasonable to assume this distribution of scores to be similar to that on $\mathcal{S}$. The latter is an "ideal" operating condition of training on labeled data and running inference on within-domain images. Let the actual distribution of baseline detector scores on $\mathcal{T}$ have p.d.f. $f(x)$, and the distribution of scores on $\mathcal{S}$ have p.d.f. $g(x)$. Let their cumulative distributions be $F(x) = \int_0^x f(t)dt$ and $G(x) = \int_0^x g(r)dr$, respectively. As a parameter-free method of creating soft-labels for our pseudo-labels on $\mathcal{T}$, we can use histogram specification [18] to map the baseline detector scores on $\mathcal{T}$ to match the distribution of scores on images from $\mathcal{S}$, *i.e.* replace each target domain score x with $G^{-1}(F(x))$. The inverse mapping is done through linear interpolation. Fig. 3(a) shows the distribution of scores for a model trained on labeled WIDER-



(a) (b) (c)

Figure 3: **Cross-domain score mapping.** Distribution of high-confidence detection scores of a face detector trained on labeled images from WIDER-Face [67]; samples are from (a) WIDER-validation and (b) CS6 surveillance videos [29]; (c) remapping the scores on CS6 to resemble WIDER.

Face [67] and run on images from the validation split of the same dataset. In Fig. 3(b), due to the domain shift, there is a visible difference when this model is run on unlabeled images from CS6 surveillance videos [29]. Fig. 3(c) shows the effect of histogram matching. Concretely, detector samples get soft-label $G^{-1}(F(d_i))$, while tracker-only samples get soft-label θ.

4. Experiments

The datasets are introduced in Sec. 4.1, followed by describing baselines (Sec. 4.2) and implementation details (Sec. 4.3). Results are shown on faces (Sec. 4.4) and pedestrians (Sec. 4.5).

4.1. Datasets

Experiments are performed on two challenging scenarios – pedestrian detection from driving videos and face detection from surveillance videos, both of which fit neatly into our paradigm of self-training from large amounts of unlabeled videos and where there exists a significant domain shift between source and target. Several example images are shown in Fig. 1. We select single-category detection tasks like face and pedestrian to avoid the engineering and computational burden of handling multiple categories, and focus on the unsupervised domain adaptation aspect. A summary of the datasets is given in Table 1.

Face: WIDER $\rightarrow$ CS6. The WIDER dataset [67] is the the source domain, consisting of labeled faces in still images downloaded from the internet with a wide variety of scale, pose and occlusion. The baseline detector is trained on the WIDER Train split, which has 12,880 images and 159,424 annotated faces. The target domain consists of 179 surveillance videos from CS6, which is a subset of the IJB-S benchmark [29]. CS6 provides a considerable shift from WIDER, with faces being mostly low-resolution and often occluded, and the imagery being of low picture quality, suffering from camera shake and motion blurs. The video clips are on average of 5 minutes at 30 fps, with some exceptionally long clips running for over an hour. We selected 86 videos to form the unlabeled target train set (*CS6-Train*). A

Table 1: **Dataset summary.** Details of the source and target datasets for face and pedestrian detection tasks are summarized here. N.B.– for the unlabeled target train sets, the #images and #annotations are unknown.

Dataset	**# images**	**# annots**	**# videos**
WIDER	12,880	159,424	-
CS6-Train	-	-	86
CS6-Test	70,022	217,827	80
BDD-Source	12,477	16,784	12,477
BDD-Target-Train	-	-	18,000
BDD-Target-Test	8,236	10,814	8,236

test set of 80 labeled videos, containing about 70,022 images and 217,827 face annotations, is used to evaluate the performance of the methods (*CS6-Test*).

Pedestrian: BDD(*clear,daytime*) $\rightarrow$ BDD(*rest*). The Berkeley Deep Drive 100k (BDD-100k) dataset [68] consists of 100,000 driving videos from a wide variety of scenes, weather conditions and time of day, creating a challenging and realistic scenario for domain adaptation. Each video clip is of 40 seconds duration at 30 fps; one frame out of every video is manually annotated. The source domain consists of clear, daytime conditions (*BDD(clear,daytime)*) and the target domain consists of all other conditions including night-time, rainy, cloudy, *etc*. (*BDD(rest)*). There are 12,477 labeled images forming *BDD-Source-Train*, containing 217k pedestrian annotations. We use 18k videos as the unlabeled *BDD-Target-Train* set, having approximately 21.6 million video frames (not all of which would contain pedestrians, naturally). The *BDD-Target-Test* set is comprised of 8,236 labeled images with 16,784 pedestrian annotations from *BDD(rest)*.

4.2. Baselines and Ablations

We consider the following methods as our baselines:

Baseline source. Detector trained on only the labeled source data – WIDER for faces and BDD(*clear,daytime*) for pedestrians.

Pseudo-labels from detections. High-confidence detections on the target training set are considered as training labels, followed by joint re-training of the baseline source detector. This is the naive baseline for acquiring pseudo-labels, denoted as `Det` in the results tables.

Pseudo-labels from tracking. Incorporating temporal consistency using a tracker and adding them into the set of pseudo-labels was referred to as *"Hard Positives"* by Jin *et al*. [26]; we adopt their nomenclature and refer to this as `HP`. As an ablation, we exclude detector results and keep just the *tracker-only* pseudo-labels for training (`Track`). Table 2 summarizes the details of the automatically gathered pseudo-labels. Note that using temporal constraints (`HP`) removes spurious isolated detections in addition to adding

Table 2: **Pseudo-labels summary.** Listing the number of images and object annotations obtained on the unlabeled *CS6-Train* and *BDD-Target-Train* videos. All the pseudo-labels obtained from the CS6 videos are used in re-training. For BDD, due to the massive number of videos, 100K frames were sub-sampled to form the training set.

Method	# images	# annots
CS6-Det	38,514	109,314
CS6-HP	15,092	84,662
CS6-Track	15,092	32,711
BDD-Det	100,001	205,336
BDD-Track	100,001	222,755
BDD-HP	100,001	362,936

missed objects, resulting in an overall decrease in data when compared to *Det* for CS6.

Soft labels for distillation. The *label interpolation* method as detailed in Sec. 3.3 is denoted as `Label-smooth`, and we show the effect of varying λ on the validation set. Cross-domain score distribution mapping is referred to as `score-remap` and constrained hard examples as `HP`-*cons* in the results tables.

Domain adversarial Faster-RCNN. While there are several domain adversarial methods such as ADDA [60] and CyCADA [22] for object *recognition*, we select Chen *et al*. [8] as the only method, to our knowledge, that has been integrated into the Faster R-CNN *detector*. Chen *et al*. [8] formulate the adversarial domain discriminator [15] with three separate losses – (i) predicting the domain label from the convolutional features (pre-ROI-pooling) of the entire image; (ii) predicting the domain label from the feature-representation of each proposed ROI; (iii) a consistency term between the image-level and ROI-level predictions. The region-proposals for the ROI-level loss are obtained from the Region Proposal Network (RPN) branch of the Faster R-CNN. In our experiments, we denote these models as – `DA`-*im* which applies the domain discriminator at the image level and `DA`-*im-roi*, which additionally has the instance-level discriminator and consistency term.

4.3. Training and Evaluation

We use the standard Faster R-CNN detector [45] for all our experiments, from a PyTorch implementation of the Detectron framework [17]. An ImageNet-pre-trained ResNet-50 network is used as a backbone, with ROI-Align region pooling. For faces, the baseline is trained for 80k iterations, starting from a learning rate of 0.001, dropping to 0.0001 at 50k, using 4 GPUs and a batch-size of 512. For pedestrians, the baseline was trained for 70k iterations, starting with a learning rate of 0.001 and dropping to 0.0001 at 50k. During training, face images were resized to be 800 pixels and pedestrian images were resized to be 500 pixels on the smaller side. Re-training for the target domain is always done jointly, using a single GPU – a training mini-batch is formed with samples from a labeled source image and a pseudo-labeled target image. In practice, we sample images alternately from source and target, fix 64 regions to be sampled from each image, and accumulate gradients over the two images before updating the model parameters. Domain adversarial models were implemented following Chen *et al*. [8], keeping their default hyper-parameter values.

Since unsupervised learning considers no labels *at all* on the target domain, we cannot set hyper-parameters or do best model selection based on a labeled validation set. The re-training for *all* the face models were stopped at the 10k iteration, while *all* the pedestrian models were stopped at the 30k iteration. For evaluating performance, to account for stochasticity in the training procedure, we do 5 rounds of training and evaluate each model on the labeled images from the test set. We use the MS-COCO toolkit as a consistent evaluation metric for both face and pedestrian detection, reporting Average Precision (AP) at an IoU threshold of 0.5.

4.4. Face detection results

The results on adapting from labeled WIDER Faces still-images to unlabeled CS6 surveillance video imagery are shown in Table 3.

Effect of pseudo-labels. The *baseline* detector, trained on WIDER Face, gets an AP of 15.66 on CS6-Test, which underscores the domain shift between WIDER and the surveillance video domain. Using only the high-confidence detections (θ=0.5) as training samples, *CS6*-`Det`, boosts performance to 17.29 AP. Using only samples from the tracker and ignoring all pseudo-labels from the detector, *CS6*-`Track`, brings down the performance to 11.73 AP. This can be partly attributed to the fact that we may miss a lot of actual faces in an image if we choose to train only on faces picked up by tracking alone. The combination of both tracking and detection results for training, *CS6*-`HP`, gives slightly better performance of 17.31 AP. This is a significant boost over the model trained on WIDER-Face: 15.66 $\rightarrow$ 17.31.

Effect of soft-labels. Incorporating soft target labels gives a consistent gain over the default hard labels, as seen in the `Label-smooth` numbers in Table 3. The effect of varying the distillation weight λ results in some fluctuation in performance – $\text{AP}_{\lambda=0.3}$ is 19.89, $\text{AP}_{\lambda=0.5}$ is 19.56 and $\text{AP}_{\lambda=0.7}$ is 20.80. Using the completely parameter-free methods we get 19.12 from score histogram remapping (`score-remap`) and a slightly higher number, 20.65, from `HP`-*cons*. Both are comparable to distillation with $\lambda = 0.7$.

Comparison to domain discriminator. The domain adversarial method (DA) gives a high performance on CS6 Test with an AP of 21.02 at the image-level (`DA`-*im*) and 22.18 with the instance-level adaptation included (`DA`-*im-*

Figure 4: **Qualitative results**(best zoomed-in)**.** (a) Baseline; (b) HP [26]; (c) *Ours*; (d) DA[8]. The domain adapted methods pick up prominent objects missed by the baseline (*cols 1,3-5*). On pedestrians (*cols 3-5*) the detection scores from DA is usually lower than our models', leading to lower overall performance despite correct localization.

roi). Our best numbers (20.80, 20.65) are comparable to this, given the variance over 5 rounds of training.

Table 3: **WIDER** $\rightarrow$ **CS6.** Average precision (AP) on of the CS6 surveillance videos, reported as mean and standard deviation over 5 rounds of training.

Method	**AP** (mean $\pm$ std)
Baseline: WIDER	15.66 ± 0.00
CS6-Det	17.29 ± 0.85
CS6-Track	11.73 ± 0.77
CS6-HP [26]	17.31 ± 0.60
CS6-Label-smooth($\lambda = 0.3$)	19.89 ± 0.92
CS6-Label-smooth($\lambda = 0.5$)	19.56 ± 1.53
CS6-Label-smooth($\lambda = 0.7$)	$\mathbf{20.80 \pm 1.34}$
Ours: CS6-score-remap	19.12 ± 1.29
Ours: CS6-HP-*cons*	$\mathbf{20.65 \pm 1.62}$
CS6-DA-*im* [8]	21.02 ± 0.96
CS6-DA-*im-roi* [8]	$\mathbf{22.18 \pm 1.20}$

4.5. Pedestrian detection results

The results on adapting from BDD-Source images from clear, daytime videos to unconstrained settings in BDD-Target are shown in Table 4. In addition to a new task, the target domain of BDD-Pedestrians provides a more challenging situation than CS6. The target domain now consists of multiple modes of appearance – snowy, rainy, cloudy, night-time, dusk, *etc*.; and various combinations thereof.

Effect of pseudo-labels. The *baseline* model gets a fairly low AP of 15.21, which is reasonable given the large domain shift from source to target. *BDD*-`Det`, which involves training with only the high-confidence detections (threshold $\theta = 0.8$), improves significantly over the baseline with an AP of 26.16. Using only the tracker results as pseudo-labels, *BDD*-`Track`, gives similar performance (26.28). *BDD*-`HP`, which combines pseudo-labels from both detection and tracking, gives the best performance among these (27.11). This is a significant boost over the baseline: 15.21 $\rightarrow$ 27.11.

Effect of soft-labels. Using soft labels via `Label-smooth` improves results further (27.11 $\rightarrow$ 28.59), with performance fluctuating slightly with different values of the λ hyper-parameter – $\text{AP}_{\lambda=0.3}$ is 28.59, $\text{AP}_{\lambda=0.5}$ is 28.38 and $\text{AP}_{\lambda=0.7}$ is 28.47. Creating soft-labels via score histogram matching (`score-remap`), we get an AP of 28.02. Emphasizing tracker-only samples while constraining identical behaviour on detector training samples (`HP`-*cons*) gives 28.43. Again, both these methods are comparable in performance to using `Label-smooth`, with the advantage of not having to set the λ hyper-parameter.

Comparison to domain discriminator. Adapting to the BDD-Target domain was challenging for the domain adversarial (DA) models [8], most likely due to the multiple complex appearance changes, unlike the WIDER$\rightarrow$CS6 shift which has a more homogeneous target domain. The image-level adaptation (`DA`-*im*) models gave 23.65 AP – a significant improvement over the baseline AP of 15.21. We had difficulties getting the `DA`-*im-roi* model to converge during training. Using the pseudo-labels from BDD-HP for class-balanced sampling of the ROIs during training had a stabilizing effect (denoted by BDD-DA-*im-roi**). This gives 23.69 AP. Overall our results from training with soft pseudo-labels are better than [8] on this dataset by ∼5 in terms of AP.

Table 4: **BDD(*clear,daytime*) → BDD(*rest*).** Average precision (AP) on the evaluation set of the BDD pedestrian videos, reported as mean and standard deviation over 5 rounds of training.

Method	**AP** (mean ± std)
Baseline: BDD(*clear,daytime*)	15.21 ± 0.00
BDD-Det	26.16 ± 0.24
BDD-Track	26.28 ± 0.35
BDD-HP [26]	27.11 ± 0.54
BDD-Label-smooth($\lambda = 0.3$)	**28.59 ± 0.67**
BDD-Label-smooth($\lambda = 0.5$)	28.38 ± 0.62
BDD-Label-smooth($\lambda = 0.7$)	28.47 ± 0.41
Ours: BDD-score-remap	*28.02 ± 0.32*
Ours: BDD-HP-*cons*	**28.43 ± 0.51**
BDD-DA-*im* [8]	23.65 ± 0.57
BDD-DA-*im-roi**	**23.69 ± 0.93**

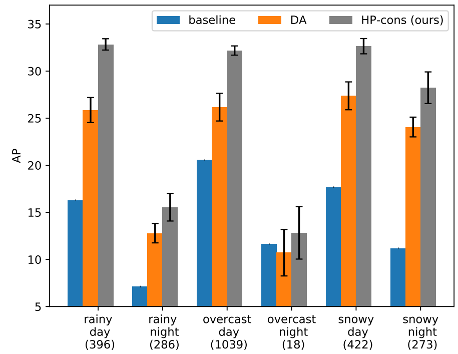


Figure 5: **BDD(*rest*) sub-domains.** Performance of the *baseline* model, domain adversarial model (*DA*) and our method (*HP-cons*). The number of images in each sub-domain is written in parentheses below.

Table 5: Sensitivity to detector confidence threshold for target-domain pseudo-labels, evaluated for the *HP* model. The automatically selected thresholds $\theta_{\mathcal{S}\to\mathcal{T}}$ are 0.66 for CS6 and 0.81 for BDD.

$\theta \rightarrow$	0.5	0.6	0.7	0.8	0.9	$\theta_{\mathcal{S}\to\mathcal{T}}$
CS6-Test	17.31	15.91	14.93	15.63	11.69	*16.71*
BDD-Test	27.23	27.68	27.30	27.11	25.85	*27.11*

Results on sub-domains. The BDD-Target domain implicitly contains a large number of *sub-domains* such as rainy, foggy, night-time, dusk, *etc*. We compare the performance of three representative models – baseline, domain adversarial (DA-*im*) and our soft-labeling method (we pick HP-*cons* as representative) on a set of such implicit sub-domains in BDD-Target-Test for a fine-grained performance analysis (Fig. 5). Night-time images clearly degrade performance for all the models. Overall both domain adaptive methods improve significantly over the baseline, with HP-*cons* consistently outperforming DA. It is possible that higher performance from DA can be obtained by some dataset-specific tuning of hyper-parameters on a validation set of *labeled* target-domain data.

4.6. Automatic threshold selection

The hyper-parameter θ that thresholds the high-confidence detections can be set without manual inspection of the target domain. We can pick a threshold $\theta_{\mathcal{S}}$ on *labeled source* data for a desired level of precision, say 0.95. Using score histogram mapping $\mathcal{S} \rightarrow \mathcal{T}$ (Sec 3.3, Fig. 3), we can map $\theta_{\mathcal{S}}$ to the *unlabeled target* domain as $\theta_{\mathcal{T}}$. These results are shown in Table 5. The thresholds selected based on visual inspection of 5 videos are 0.5 for faces (17.31 AP) and 0.8 for pedestrians (27.11 AP), as described in Sec. 3.1. The performance from automatically set $\theta_{\mathcal{S}\to\mathcal{T}}$ is very close – AP of 16.71 on CS6 and 27.11 on BDD.

5. Conclusion

Our empirical analysis shows self-training with soft-labels to be at par with or better than the recent domain adversarial approach [8] on two challenging tasks. Our method also avoids the extra layers and hyper-parameters of adversarial methods, which are difficult to tune for novel domains in a fully unsupervised scenario. Our method significantly boosts the performance of pre-trained models on the target domain and gives a consistent improvement over assigning hard labels to pseudo-labeled target domain samples, the latter being prevalent in recent works [26, 43]. With minimal dependence on hyper-parameters, we believe our approach to be a readily applicable method for large-scale domain adaptation of object detectors.

Acknowledgement. This material is based on research sponsored by the AFRL and DARPA under agreement number FA8750-18-2-0126. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the AFRL and DARPA or the U.S. Government. We acknowledge support from the MassTech Collaborative grant for funding the UMass GPU cluster. We thank Tsung-Yu Lin and Subhransu Maji for helpful discussions.

References

[1] M. Abdullah Jamal, H. Li, and B. Gong. Deep face detector adaptation without negative transfer or catastrophic forgetting. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 3

[2] S. Baluja. Probabilistic modeling for face orientation discrimination: Learning from labeled and unlabeled data. In *Advances in Neural Information Processing Systems*, pages 854–860, 1999. 2

[3] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan. A theory of learning from different domains. *Machine learning*, 79(1-2):151–175, 2010. 2

[4] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 92–100. ACM, 1998. 2

[5] K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, and D. Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, page 7, 2017. 2

[6] O. Chanda, E. W. Teh, M. Rochan, Z. Guo, and Y. Wang. Adapting object detectors from images to weakly labeled videos. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2017. 3

[7] O. Chapelle, B. Scholkopf, and A. Zien. Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. *IEEE Transactions on Neural Networks*, 20(3):542–542, 2009. 2

[8] Y. Chen, W. Li, C. Sakaridis, D. Dai, and L. Van Gool. Domain adaptive faster r-cnn for object detection in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3339–3348, 2018. 2, 3, 6, 7, 8

[9] G. Csurka. Domain adaptation for visual applications: A comprehensive survey. *arXiv preprint arXiv:1702.05374*, 2017. 2

[10] D. Dai, C. Sakaridis, S. Hecker, and L. Van Gool. Curriculum model adaptation with synthetic and real data for semantic foggy scene understanding. *arXiv preprint arXiv:1901.01415*, 2019. 3

[11] J. Donahue, J. Hoffman, E. Rodner, K. Saenko, and T. Darrell. Semi-supervised domain adaptation with instance constraints. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 668–675, 2013. 3

[12] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, June 2010. 1

[13] C. Feichtenhofer, A. Pinz, and A. Zisserman. Detect to track and track to detect. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3038–3046, 2017. 2

[14] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 2, pages II–II. IEEE, 2003. 2

[15] Y. Ganin and V. Lempitsky. Unsupervised domain adaptation by backpropagation. *arXiv preprint arXiv:1409.7495*, 2014. 2, 3, 6

[16] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030, 2016. 2, 3

[17] R. Girshick, I. Radosavovic, G. Gkioxari, P. Dollár, and K. He. Detectron. https://github.com/facebookresearch/detectron, 2018. 1, 6

[18] R. C. Gonzalez, R. E. Woods, et al. Digital image processing, 2002. 4

[19] A. Gretton, A. Smola, J. Huang, M. Schmittfull, K. Borgwardt, and B. Schölkopf. Dataset shift in machine learning. In *Covariate Shift and Local Learning by Distribution Matching*, pages 131–160. MIT Press, 2008. 2

[20] T. Han, G. Hua, and X. Wang. Detection by detections: Non-parametric detector adaptation for a video. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 350–357. IEEE, 2012. 3

[21] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 4

[22] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. A. Efros, and T. Darrell. Cycada: Cycle-consistent adversarial domain adaptation. *arXiv preprint arXiv:1711.03213*, 2017. 2, 6

[23] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, et al. Speed/accuracy trade-offs for modern convolutional object detectors. In *IEEE CVPR*, volume 4, 2017. 1

[24] N. Inoue, R. Furuta, T. Yamasaki, and K. Aizawa. Cross-domain weakly-supervised object detection through progressive domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5001–5009, 2018. 2, 3

[25] Y. Jiang and Z.-H. Zhou. Editing training data for knn classifiers with neural network ensemble. In *International symposium on neural networks*, pages 356–361. Springer, 2004. 2

[26] S. Jin, A. RoyChowdhury, H. Jiang, A. Singh, A. Prasad, D. Chakraborty, and E. Learned-Miller. Unsupervised hard example mining from videos for improved object detection. In *European Conference on Computer Vision (ECCV)*, 2018. 2, 3, 4, 5, 7, 8

[27] S. Jin, H. Su, C. Stauffer, and E. Learned-Miller. End-to-end face detection and cast grouping in movies using erdos-renyi clustering. In *ICCV*, 2017. 3

[28] Z. Kalal, J. Matas, and K. Mikolajczyk. Pn learning: Bootstrapping binary classifiers by structural constraints. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 49–56. IEEE, 2010. 3

[29] N. D. Kalka, B. Maze, J. A. Duncan, K. OConnor, S. Elliott, K. Hebert, J. Bryan, and A. K. Jain. Ijb–s: Iarpa janus surveillance video benchmark. 1, 3, 5

[30] V. Kalogeiton, V. Ferrari, and C. Schmid. Analysing domain shift factors between videos and images for object detection.

IEEE Transactions on Pattern Analysis and Machine Intelligence, 38(11):2327–2334, 2016. 2

[31] A. Kuznetsova, S. Ju Hwang, B. Rosenhahn, and L. Sigal. Expanding object detector's horizon: incremental learning framework for object detection in videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 28–36, 2015. 3

[32] D.-H. Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *ICML Workshop*, volume 3, page 2, 2013. 2, 3

[33] A. Levin, P. A. Viola, and Y. Freund. Unsupervised improvement of visual detectors using co-training. In *ICCV*, volume 1, pages 626–633, 2003. 2

[34] M. Li and Z.-H. Zhou. Setred: self-training with editing. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 611–621. Springer, 2005. 2

[35] Y. Li, J. Yang, Y. Song, L. Cao, J. Luo, and L.-J. Li. Learning from noisy labels with distillation. In *ICCV*, pages 1928–1936, 2017. 4

[36] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 1

[37] M. Long, Y. Cao, J. Wang, and M. I. Jordan. Learning transferable features with deep adaptation networks. *arXiv preprint arXiv:1502.02791*, 2015. 2

[38] L. Ma, S. Tang, M. J. Black, and L. V. Gool. Customized multi-person tracker. In *Computer Vision – ACCV 2018*. Springer International Publishing, Dec. 2018. 2

[39] F. Muhlenbach, S. Lallich, and D. A. Zighed. Identifying and handling mislabelled instances. *Journal of Intelligent Information Systems*, 22(1):89–109, 2004. 2

[40] H. Nam and B. Han. Learning multi-domain convolutional neural networks for visual tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4293–4302, 2016. 3

[41] K. Nigam and R. Ghani. Analyzing the effectiveness and applicability of co-training. In *Proceedings of the ninth international conference on Information and knowledge management*, pages 86–93. ACM, 2000. 2

[42] A. Odena, A. Oliver, C. Raffel, E. D. Cubuk, and I. Goodfellow. Realistic evaluation of semi-supervised learning algorithms. 2018. 2

[43] I. Radosavovic, P. Dollár, R. Girshick, G. Gkioxari, and K. He. Data distillation: Towards omni-supervised learning. *arXiv preprint arXiv:1712.04440*, 2017. 2, 8

[44] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2016. 3

[45] S. Ren, K. He, R. B. Girshick, and J. Sun. Faster R-CNN: towards real-time object detection with region proposal networks. In *NIPS*, pages 91–99, 2015. 3, 6

[46] C. Rosenberg, M. Hebert, and H. Schneiderman. Semi-supervised self-training of object detection models. 2005. 2

[47] C. Sakaridis, D. Dai, S. Hecker, and L. Van Gool. Model adaptation with synthetic and real data for semantic dense foggy scene understanding. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 687–704, 2018. 3

[48] R. E. Schapire, Y. Freund, P. Bartlett, W. S. Lee, et al. Boosting the margin: A new explanation for the effectiveness of voting methods. *The annals of statistics*, 26(5):1651–1686, 1998. 2

[49] A. Selinger and R. C. Nelson. Minimally supervised acquisition of 3d recognition models from cluttered images. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–I. IEEE, 2001. 2

[50] P. Sharma and R. Nevatia. Efficient detector adaptation for object detection in a video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3254–3261, 2013. 3

[51] A. Shrivastava, A. Gupta, and R. Girshick. Training region-based object detectors with online hard example mining. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 761–769, 2016. 2, 3

[52] K. K. Singh, F. Xiao, and Y. J. Lee. Track and transfer: Watching videos to simulate strong human supervision for weakly-supervised object detection. In *CVPR*, volume 1, page 2, 2016. 3

[53] M. Sugiyama, N. D. Lawrence, A. Schwaighofer, et al. *Dataset shift in machine learning*. The MIT Press, 2017. 2

[54] B. Sun and K. Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *European Conference on Computer Vision*, pages 443–450. Springer, 2016. 2

[55] K.-K. Sung and T. Poggio. Learning and example selection for object and pattern detection. 1994. 3

[56] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016. 4

[57] K. Tang, V. Ramanathan, L. Fei-Fei, and D. Koller. Shifting weights: Adapting object detectors from image to video. In *Advances in Neural Information Processing Systems*, pages 638–646, 2012. 3

[58] I. Triguero, S. García, and F. Herrera. Self-labeled techniques for semi-supervised learning: taxonomy, software and empirical study. *Knowledge and Information Systems*, 42(2):245–284, 2015. 3

[59] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko. Simultaneous deep transfer across domains and tasks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4068–4076, 2015. 2, 4

[60] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell. Adversarial discriminative domain adaptation. In *Computer Vision and Pattern Recognition (CVPR)*, volume 1, page 4, 2017. 2, 6

[61] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*, 2014. 2

[62] S. Wang, Y. Zhou, J. Yan, and Z. Deng. Fully motion-aware network for video object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 542–557, 2018. 2

[63] M. Weber, M. Welling, and P. Perona. Unsupervised learning of models for recognition. In *European conference on computer vision*, pages 18–32. Springer, 2000. 2

[64] D. Weinshall and D. Amir. Theory of curriculum learning, with convex loss functions. *arXiv preprint arXiv:1812.03472*, 2018. 2

[65] J. WESTON. Large-scale semi-supervised learning. 2

[66] D. L. Wilson. Asymptotic properties of nearest neighbor rules using edited data. *IEEE Transactions on Systems, Man, and Cybernetics*, (3):408–421, 1972. 2

[67] S. Yang, P. Luo, C. C. Loy, and X. Tang. WIDER FACE: A face detection benchmark. In *CVPR*, 2016. 1, 5

[68] F. Yu, W. Xian, Y. Chen, F. Liu, M. Liao, V. Madhavan, and T. Darrell. Bdd100k: A diverse driving video database with scalable annotation tooling. *arXiv preprint arXiv:1805.04687*, 2018. 1, 5

[69] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV)*, 2017. 2, 3