This ICCV paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

A Robust Learning Approach to Domain Adaptive Object Detection

Mehran Khodabandeh
Simon Fraser University
Burnaby, Canada
mkhodaba@sfu.ca

Arash Vahdat
NVIDIA
California, USA
avahdat@nvidia.com

Mani Ranjbar
Quadrant
Burnaby, Canada
mani@quadrant.ai

William G. Macready
Quadrant
Burnaby, Canada
bill@quadrant.ai

Abstract

Domain shift is unavoidable in real-world applications of object detection. For example, in self-driving cars, the target domain consists of unconstrained road environments which cannot all possibly be observed in training data. Similarly, in surveillance applications sufficiently representative training data may be lacking due to privacy regulations. In this paper, we address the domain adaptation problem from the perspective of robust learning and show that the problem may be formulated as training with noisy labels. We propose a robust object detection framework that is resilient to noise in bounding box class labels, locations and size annotations. To adapt to the domain shift, the model is trained on the target domain using a set of noisy object bounding boxes that are obtained by a detection model trained only in the source domain. We evaluate the accuracy of our approach in various source/target domain pairs and demonstrate that the model significantly improves the state-of-the-art on multiple domain adaptation scenarios on the SIM10K, Cityscapes and KITTI datasets.

1. Introduction

Object detection lies at the core of computer vision and finds application in surveillance, medical imaging, self-driving cars, face analysis, and industrial manufacturing. Recent advances in object detection using convolutional neural networks (CNNs) have made current models fast, reliable and accurate.

However, domain adaptation remains a significant challenge in object detection. In many discriminative problems (including object detection) it is usually assumed that the distribution of instances in both train (source domain) and test (target domain) set are identical. Unfortunately, this assumption is easily violated, and domain changes in object detection arise with variations in viewpoint, background, object appearance, scene type and illumination. Further, object detection models are often deployed in environments which differ from the training environment.

Common domain adaptation approaches are based on either supervised model fine-tuning in the target domain or unsupervised cross-domain representation learning. While the former requires additional labeled instances in the target domain, the latter eliminates this requirement at the cost of two new challenges. Firstly, the source/target representations should be matched in some space (e.g., either in input space [69, 23] or hidden representations space [14, 54]). Secondly, a mechanism for feature matching must be defined (*e.g.* maximum mean discrepancy (MMD) [40, 35], $\mathcal{H}$ divergence [2], or adversarial learning).

In this paper, we approach domain adaptation differently, and address the problem through robust training methods. Our approach relies on the observation that, although a (primary) model trained in the source domain may have suboptimal performance in the target domain, it may nevertheless be used to detect objects in the target domain with some accuracy. The detected objects can then be used to retrain a detection model on both domains. However, because the instances detected in the target domain may be inaccurate, a robust detection framework (which accommodates these inaccuracies) must be used during retraining.

The principal benefit of this formulation is that the detection model is trained in an unsupervised manner in the target domain. Although we do not explicitly aim at matching representations between source and target domain, the detection model may implicitly achieve this because it is fed by instances from both source and target domains.

To accommodate labeling inaccuracies we adopt a probabilistic perspective and develop a robust training framework for object detection on top of Faster R-CNN [47]. We provide robustness against two types of noise: i) mistakes in object labels (*i.e.*, a bounding box is labeled as person but actually is a pole), and ii) inaccurate bounding box location and size (*i.e.*, a bounding box does not enclose the object). We formulate the robust retraining objective so that the model can alter both bounding box class labels and bounding box location/size based on its current belief of labels in the target domain. This enables the robust detection

model to refine the noisy labels in the target domain.

To further improve label quality in the target domain, we introduce an auxiliary image classification model. We expect that an auxiliary classifier can improve target domain labels because it may use cues that have not been utilized by the original detection model. As examples, additional cues can be based on additional input data (*e.g.* motion or optical flow), different network architectures, or ensembles of models. We note however, that the auxiliary image classification model is only used during the retraining phase and the computational complexity of the final detector is preserved at test time.

The contributions of this paper are summarized as follows: i) We provide the first (to the best of our knowledge) formulation of domain adaptation in object detection as robust learning. ii) We propose a novel robust object detection framework that considers noise in training data on both object labels and locations. We use Faster R-CNN[47] as our base object detector, but our general framework, theoretically, could be adapted to other detectors (e.g. SSD [32] and YOLO [45]) that minimize a classification loss and regress bounding boxes. iii) We use an independent classification refinement module to allow other sources of information from the target domain (*e.g.* motion, geometry, background information) to be integrated seamlessly. iv) We demonstrate that this robust framework achieves state-of-the-art on several cross-domain detection tasks.

2. Previous Work

Object Detection: The first approaches to object detection used a sliding window followed by a classifier based on hand-crafted features [6, 11, 60]. After advances in deep convolutional neural networks, methods such as R-CNN [19], SPPNet [22], and Fast R-CNN [18] arose which used CNNs for feature extraction and classification. Slow sliding window algorithms were replaced with faster region proposal methods such as selective search [55]. Recent object detection methods further speed bounding box detection. For example, in Faster R-CNN [47] a region proposal network (RPN) was introduced to predict refinements in the locations and sizes of predefined anchor boxes. In SSD [32], classification and bounding box prediction is performed on feature maps at different scales using anchor boxes with different aspect ratios. In YOLO [44], a regression problem on a grid is solved, where for each cell in the grid, the bounding box and the class label of the object centering at that cell is predicted. Newer extensions are found in [66, 45, 5]. A comprehensive comparison of methods is reported in [25]. The goal of this paper is to increase the accuracy of an object detector in a new domain regardless of the speed. Consequently, we base our improvements on Faster R-CNN, a slower, but accurate detector.[1]

Domain Adaptation: was initially studied for image classification and the majority of the domain adaptation literature focuses on this problem [10, 9, 30, 21, 20, 12, 50, 33, 34, 14, 13, 17, 1, 39, 31]. Some of the methods developed in this context include cross-domain kernel learning methods such as adaptive multiple kernel learning (A-MKL) [10], domain transfer multiple kernel learning (DTMKL) [9], and geodesic flow kernel (GFK) [20]. There are a wide variety of approaches directed towards obtaining domain invariant predictors: supervised learning of non-linear transformations between domains using asymmetric metric learning [30], unsupervised learning of intermediate representations [21], alignment of target and domain subspaces using eigenvector covariances [12], alignment the second-order statistics to minimize the shift between domains [50], and covariance matrix alignment approach [62]. The rise of deep learning brought with it steps towards domain-invariant feature learning. In [33, 34] a reproducing kernel Hilbert embedding of the hidden features in the network is learned and mean-embedding matching is performed for both domain distributions. In [14, 13] an adversarial loss along with a domain classifier is trained to learn features that are discriminative and domain invariant.

There is less work in domain adaptation for object detection. Domain adaptation methods for non-image classification tasks include [15] for fine-grained recognition, [3, 24, 67, 61] for semantic segmentation, [29] for dataset generation, and [36] for finding out of distribution data in active learning. For object detection itself, [64] used an adaptive SVM to reduce the domain shift, [43] performed subspace alignment on the features extracted from R-CNN, and [2] used Faster RCNN as baseline and took an adversarial approach (similar to [13]) to learn domain invariant features jointly on target and source domains. We take a fundamentally different approach by reformulating the problem as noisy labeling. We design a robust-to-noise training scheme for object detection which is trained on noisy bounding boxes and labels acquired from the target domain as pseudo-ground-truth.

Noisy Labeling: Previous work on robust learning has focused on image classification where there are few and disjoint classes. Early work used instance-independent noise models, where each class is confused with other classes independent of the instance content [41, 38, 42, 49, 68, 65]. Recently, the literature has shifted towards instance-specific label noise prediction [63, 37, 56, 57, 58, 59, 53, 27, 7, 46]. To the best of our knowledge, ours is the first proposal for an object detection model that is robust to label noise.

[1] Our adoption of faster R-CNN also allows for direct comparison with the state-of-the-art [2].

3. Method

Following the common formulation for domain adaptation, we represent the training data space as the source domain ($\mathcal{S}$) and the test data space as the target domain ($\mathcal{T}$). We assume that an annotated training image dataset in $\mathcal{S}$ is supplied, but that only images in $\mathcal{T}$ are given (*i.e.* there are no labels in $\mathcal{T}$). Our framework, visualized in Fig. 1, consists of three main phases:

1. **Object proposal mining**: A standard Faster R-CNN, trained on the source domain, is used to detect objects in the target domain. The detected objects form a proposal set in $\mathcal{T}$.
2. **Image classification training**: Given the images extracted from bounding boxes in $\mathcal{S}$, we train an image classification model that predicts the class of objects in each image. The resulting classifier is used to score the proposed bounding boxes in $\mathcal{T}$. This model aids in training the robust object detection model in the next phase. The reason for introducing image classification is that i) this model may rely on representations different than those used by the phase one detection model (*e.g.*, motion features) or it may use a more sophisticated network architectures, and ii) this model can be trained in a semi-supervised fashion using labeled images in $\mathcal{S}$ and unlabeled images in $\mathcal{T}$.
3. **Robust object detection training**: In this phase a robust object detection model is trained using object bounding boxes in $\mathcal{S}$ and object proposals in $\mathcal{T}$ (from phase one) that has been rescored using the image classification (from phase two).

We organize the detailed method description as follows. Firstly, we introduce background notation and provide a description of Faster R-CNN in Sec. 3.1 to define the model used in phase one. Secondly, a probabilistic view of Faster R-CNN in Sec. 3.2 provides a foundation for the robust object detection framework presented in Sec. 3.3. This defines the model used in phase three. Lastly, the image classification model used in phase two is discussed in Sec. 3.4.

Notation: We are given training images in $\mathcal{S}$ along with their object bounding box labels. This training set is denoted by $\boldsymbol{D}_{\mathcal{S}} = \{(\boldsymbol{x}^{(s)}, \boldsymbol{y}^{(s)})\}$ where $\boldsymbol{x}^{(s)} \in \mathcal{S}$ represents an image, $\boldsymbol{y}^{(s)}$ is the corresponding bounding box label for $\boldsymbol{x}^{(s)}$ and s is an index. Each bounding box $\boldsymbol{y} = (y_c, \boldsymbol{y}_l)$ represents a class label by an integer, $y_c \in \mathcal{Y} = \{1, 2, \ldots, C\}$, where C is the number of foreground classes, and a 4-tuple, $\boldsymbol{y}_l \in \mathcal{R}^4$, giving the coordinates of the top left corner, height, and width of the box. To simplify notation, we associate each image with a single bounding box.[2]

[2]This restriction is for notational convenience only. Our implementation makes no assumptions about the number of objects in each image.

In the target domain, images are given without bounding box annotations. At the end of phase one, we augment this dataset with proposed bounding boxes generated by Faster R-CNN. We denote the resulting set by $\boldsymbol{D}_{\mathcal{T}} = \{\boldsymbol{x}^{(t)}, \tilde{\boldsymbol{y}}^{(t)}\}$ where $\boldsymbol{x}^{(t)} \in \mathcal{T}$ is an image, $\tilde{\boldsymbol{y}}^{(t)} \in \mathcal{Y}$ is the corresponding proposed bounding box and t is an index. Finally, we obtain the image classification score obtained at the end of phase two for each instance in $\boldsymbol{D}_{\mathcal{T}}$ from $p_{img}(y_c|\boldsymbol{x}, \tilde{\boldsymbol{y}}_l)$ which represents the probability of assigning the image cropped in the bounding box $\tilde{\boldsymbol{y}}_l$ in $\boldsymbol{x}$ to the class $y_c \in \mathcal{Y} \cup \{0\}$ which is one of the foreground categories or background.

3.1. Faster R-CNN

Faster R-CNN [47] is a two-stage detector consisting of two main components: a region proposal network (RPN) that proposes regions of interests (ROI) for object detection and an ROI classifier that predicts object labels for the proposed bounding boxes. These two components share the first convolutional layers. Given an input image, the shared layers extract a feature map for the image. In the first stage, RPN predicts the probability of a set of predefined anchor boxes for being an object or background along with refinements in their sizes and locations. The anchor boxes are a fixed predefined set of boxes with varying positions, sizes and aspect ratios across the image. Similar to RPN, the region classifier predicts object labels for ROIs proposed by the RPN as well as refinements for the location and size of the boxes. Features passed to the classifier are obtained with a *ROI-pooling* layer. Both networks are trained jointly by minimizing a loss function:

$$\mathcal{L} = \mathcal{L}_{RPN} + \mathcal{L}_{ROI}. \tag{1}$$

$\mathcal{L}_{RPN}$ and $\mathcal{L}_{ROI}$ represent losses used for the RPN and ROI classifier. The losses consist of a cross-entropy cost measuring the mis-classification error and a regression loss quantifying the localization error. The RPN is trained to detect and localize objects without regard to their classes, and the ROI classification network is trained to classify the object labels.

3.2. A Probabilistic View of Faster R-CNN

In this section, we provide a probabilistic view of Faster R-CNN that will be used to define a robust loss function for noisy detection labels. The ROI classifier in Faster R-CNN generates an object classification score and object location for each proposed bounding box generated by the RPN. A classification prediction $p_{cls}(y_c|\boldsymbol{x}, \tilde{\boldsymbol{y}}_l)$ represents the probability of a categorical random variable taking one of the disjoint $C + 1$ classes (*i.e.*, foreground classes plus background). This classification distribution is modeled using a softmax activation. Similarly, we model the location prediction $p_{loc}(\boldsymbol{y}_l|\boldsymbol{x}, \tilde{\boldsymbol{y}}_l) = \mathcal{N}(\boldsymbol{y}_l; \bar{\boldsymbol{y}}_l, \sigma\boldsymbol{I})$ with a multivariate

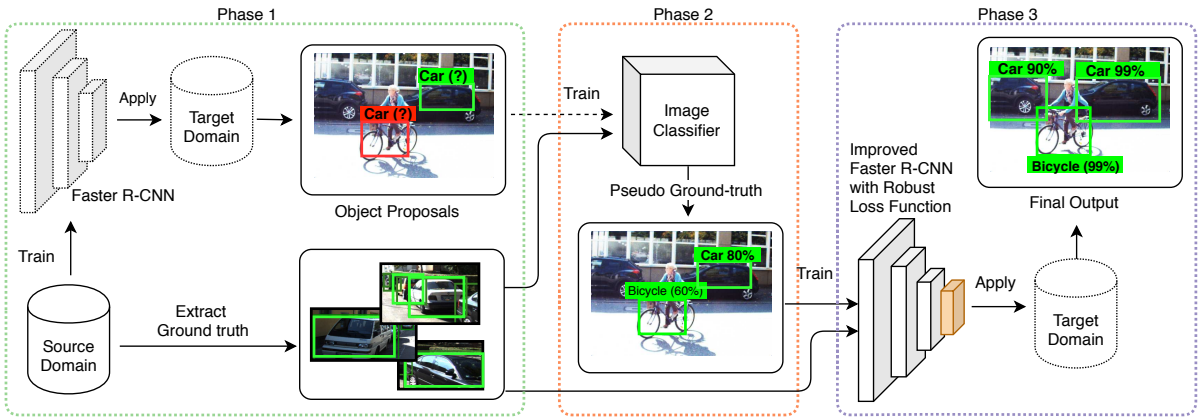

Figure 1: The robust learning approach consists of three phases. In phase 1, a detection module is trained using labeled data in the source domain. This detector is then used to generate noisy annotations for images in the target domain. In phase 2, the annotations assigned in phase 1 are refined using a classification module. Finally, in phase 3, the detector is retrained using the original labeled data and the refined machine-generated annotations in the target domain. Retraining is formulated to account for the possibility of mislabeling.

Normal distribution[3] with mean $\bar{\boldsymbol{y}}_l$ and constant diagonal covariance matrix $\sigma\boldsymbol{I}$. In practice, only $\bar{\boldsymbol{y}}_l$ is generated by the ROI classifier which is used to localize the object.

3.3. Robust Faster R-CNN

To gain robustness against detection noise on both the label (y_c) and the box location/size ($\boldsymbol{y}_l$), we develop a refinement mechanism that corrects mistakes in both class and box location/size annotations. The phase three detection model is trained using these refined annotations.

If the training annotations are assumed to be noise-free then both p_{cls} and p_{loc} are used to define the maximum-likelihood loss functions in Eq. 1. In the presence of noisy labels, $\arg\max p_{cls}$ and $\arg\max p_{loc}$ may disagree with the noisy labels but nevertheless correctly identify the true class or location of an object. Additionally, we also have access to the image classification model p_{img} from phase 2 that may be more accurate in predicting class labels for proposed bounding boxes in $\mathcal{T}$ since it is trained using information sources different from the primary detection model. The question then is how to combine p_{cls}, p_{loc} from Faster R-CNN and p_{img} from the image model to get the best prediction for the class and location of an object?

Vahdat [56] has proposed a regularized EM algorithm for robust training of image classification models. Inspired by this approach, we develop two mechanisms for correcting classification and localization errors, based on the assumption that when training a classification model on noisy labeled instances, the distribution over true labels should be close to both the distributions generated by the underlying classification model and an auxiliary distribution obtained from other sources. Since the accuracy of the learned classification model improves during training, the weighting of these information sources should shift during training.

Classification Error Correction: We seek a distribution, $q(y_c)$, which is close to both the classification model of Faster R-CNN and the image classification model p_{img}, that is trained in phase two. We propose the following optimization objective for inferring $q(y_c)$

$$\min_q \mathrm{KL}(q(y_c)||p_{cls}(y_c|\boldsymbol{x},\tilde{\boldsymbol{y}}_l))+\alpha\mathrm{KL}(q(y_c)||p_{img}(y_c|\boldsymbol{x},\tilde{\boldsymbol{y}}_l)). \tag{2}$$

KL denotes the Kullback-Leibler divergence and $\alpha > 0$ balances the trade-off between two terms. With large values of α, q favors the image classification model (p_{img}) over Faster R-CNN predictions (p_{cls}), and with smaller α, q favors p_{cls}. Over the course of training, α can be changed to set a reasonable balance between the two distributions.

The following result provides a closed-form solution to the optimization problem in Eq. 2:

Theorem 1. *Given two probability distributions $p_1(z)$ and $p_2(z)$ defined for the random variable z and positive scalar α, the closed-form minimizer of*

$$\min_q KL(q(z)||p_1(z)) + \alpha KL(q(z)||p_2(z))$$

is given by:

$$q(z) \propto \left(p_1(z)p_2^{\alpha}(z)\right)^{\frac{1}{\alpha+1}} \tag{3}$$

Proof. Here, we prove the theorem for a continuous random

[3]This assumption follows naturally if the L_2-norm is used for the localization error in Eq. 1. In practice however, a combination of L_2 and L_1 norms are used which do not correspond to a simple probabilistic output.

variable defined in domain Ω.

$$\begin{aligned}\min_q \quad & \mathrm{KL}(q(z)||p_1(z)) + \alpha \mathrm{KL}(q(z)||p_2(z)) \\ &= \int_\Omega q(z) \log \frac{q(z)}{p_1(z)} dz + \alpha \int_\Omega q(z) \log \frac{q(z)}{p_2(z)} dz \\ &= (\alpha+1) \int_\Omega q(z) \log \frac{q(z)}{\left[p_1(z) p_2^\alpha(z)\right]^{\frac{1}{\alpha+1}}} dz \\ &= (\alpha+1)\mathrm{KL}(q(z) \,||\, \frac{1}{Z}\left[p_1(z)p_2^\alpha(z)\right]^{\frac{1}{\alpha+1}}) + C\end{aligned}$$

where Z is the normalization for $\left(p_1(z)p_2^\alpha(z)\right)^{\frac{1}{\alpha+1}}$ and C is a constant independent of q. The final KL is minimized when Eq. 3 holds. □

Using Theorem. 1, the solution to Eq. 2 is obtained as the weighted geometric mean of the two distributions:

$$q(y_c) \propto \left(p_{cls}(y_c|\boldsymbol{x}, \tilde{\boldsymbol{y}}_l) p_{img}^\alpha(y_c|\boldsymbol{x}, \tilde{\boldsymbol{y}}_l)\right)^{\frac{1}{\alpha+1}}. \quad (4)$$

Since both $p_{cls}(y_c|\boldsymbol{x}, \tilde{\boldsymbol{y}}_l)$ and $p_{img}(y_c|\boldsymbol{x}, \tilde{\boldsymbol{y}}_l)$ are categorical distributions (with softmax activation), $q(y_c)$ is also a (softmax) categorical distribution whose parameters are obtained as the weighted mean of the logits generated by p_{cls} and p_{img}, *i.e.*, $\sigma\left((\boldsymbol{l}_{cls}+\alpha \boldsymbol{l}_{img})/(1+\alpha)\right)$ where σ is the softmax and $\boldsymbol{l}_{cls}$ and $\boldsymbol{l}_{img}$ are the corresponding logits. Setting $\alpha = \infty$ in Eq. 4 sets $q(y_c)$ to $p_{img}(y_c|\boldsymbol{x}, \tilde{\boldsymbol{y}}_l)$ while $\alpha = 0$ sets $q(y_c)$ to $p_{cls}(y_c|\boldsymbol{x}, \tilde{\boldsymbol{y}}_l)$. During training we reduce α from large to smaller values. Intuitively, at the beginning of the training, $p_{cls}(y_c|\boldsymbol{x}, \tilde{\boldsymbol{y}}_l)$ is inaccurate and provides a poor estimation of the true class labels, therefore by setting α to a large value we guide $q(y_c)$ to rely on $p_{img}(y_c|\boldsymbol{x}, \tilde{\boldsymbol{y}}_l)$ more than p_{cls}. By decreasing α throughout training, q will rely on both p_{cls} and p_{img} to form a distribution over true class labels.

Bounding Box Refinement: Eq. 4 refines the classification labels for the proposal bounding boxes in the target domain. Here, we provide a similar method for correcting the errors in location and size. Recall that Faster R-CNN's location predictions for the proposal bounding boxes can be thought as a Normally distributed $\mathcal{N}(\boldsymbol{y}_l; \tilde{\boldsymbol{y}}_l, \sigma \boldsymbol{I})$ with mean $\tilde{\boldsymbol{y}}_l$ and constant diagonal covariance matrix $\sigma \boldsymbol{I}$. We let $p_{init}(\boldsymbol{y}_l|\boldsymbol{x}, \tilde{\boldsymbol{y}}_l) := \mathcal{N}(\boldsymbol{y}_l; \tilde{\boldsymbol{y}}_l, \sigma \boldsymbol{I})$ denote the initial detection for image $\boldsymbol{x}$. At each iteration Faster R-CNN predicts a location for object using $p_{loc}(\boldsymbol{y}_l|\boldsymbol{x}, \tilde{\boldsymbol{y}}_l) = \mathcal{N}(\boldsymbol{y}_l; \bar{\boldsymbol{y}}_l, \sigma \boldsymbol{I})$ for image $\boldsymbol{x}$ and the proposal $\tilde{\boldsymbol{y}}_l$. We use the following objective function for inferring a distribution q over true object locations:

$$\min_q \mathrm{KL}(q(\boldsymbol{y}_l)||p_{loc}(\boldsymbol{y}_l|\boldsymbol{x}, \tilde{\boldsymbol{y}}_l)) + \alpha \mathrm{KL}(q(\boldsymbol{y}_l)||p_{init}(\boldsymbol{y}_l|\boldsymbol{x}, \tilde{\boldsymbol{y}}_l)) \quad (5)$$

As with Eq. 2, the solution to Eq. 5 is the weighted geometric mean of the two distributions.

Theorem 2. *Given two multivariate Normal distributions $p_1(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z}; \boldsymbol{\mu}_1, \boldsymbol{\Sigma})$ and $p_2(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z}; \boldsymbol{\mu}_2, \boldsymbol{\Sigma})$ with common covariance matrix $\boldsymbol{\Sigma}$ defined for the random variable $\boldsymbol{z}$ and a positive scalar α, the weighted geometric mean $q(\boldsymbol{z}) \propto \left(p_1(\boldsymbol{z})p_2^\alpha(\boldsymbol{z})\right)^{\frac{1}{\alpha+1}}$ is also Normal with mean $\left(\boldsymbol{\mu}_1 + \alpha\boldsymbol{\mu}_2\right)/(\alpha+1)$ and covariance matrix $\boldsymbol{\Sigma}$.*

Proof. By the definition of the Normal distribution, we have:

$$\begin{aligned}q(\boldsymbol{z}) &\propto \left(p_1(\boldsymbol{z})p_2^\alpha(\boldsymbol{z})\right)^{\frac{1}{\alpha+1}} \\ &\propto e^{-\frac{1}{2}\left[\frac{1}{\alpha+1}(\boldsymbol{z}-\boldsymbol{\mu}_1)^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{z}-\boldsymbol{\mu}_1)+\frac{\alpha}{\alpha+1}(\boldsymbol{z}-\boldsymbol{\mu}_2)^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{z}-\boldsymbol{\mu}_2)\right]} \\ &\propto e^{-\frac{1}{2}\left[\boldsymbol{z}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{z}-2\boldsymbol{z}^T\boldsymbol{\Sigma}^{-1}(\frac{\boldsymbol{\mu}_1+\alpha\boldsymbol{\mu}_2}{\alpha+1})\right]} \\ &\propto e^{-\frac{1}{2}\left(\boldsymbol{z}-(\frac{\boldsymbol{\mu}_1+\alpha\boldsymbol{\mu}_2}{\alpha+1})\right)^T\boldsymbol{\Sigma}^{-1}\left(\boldsymbol{z}-(\frac{\boldsymbol{\mu}_1+\alpha\boldsymbol{\mu}_2}{\alpha+1})\right)}\end{aligned}$$

Hence, $q(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z}; (\boldsymbol{\mu}_1 + \alpha\boldsymbol{\mu}_2)/(\alpha+1), \boldsymbol{\Sigma})$ □

Using Theorem. 2, the minimizer of Eq. 5 is:

$$q(\boldsymbol{y}_l) = \mathcal{N}\left(\boldsymbol{y}_l; (\bar{\boldsymbol{y}}_l + \alpha\tilde{\boldsymbol{y}}_l)/(\alpha+1), \sigma\boldsymbol{I}\right). \quad (6)$$

This result gives the refined bounding box location and size as the weighted average of box location/size extracted from phase one and the current output of Faster R-CNN. Setting $\alpha = \infty$ ignores the current output of Faster R-CNN while $\alpha = 0$ uses its output as the location. At training time, we initially set α to a large value and then gradually decrease it to smaller values. In this way, at early stages of training q relies on p_{init} because it's more accurate than the current estimation of the model, but as training progresses and p_{loc} becomes more accurate, q relies more heavily on p_{loc}.

Training Objective Function: We train a robust Faster R-CNN using $\boldsymbol{D}_\mathcal{S} \cup \boldsymbol{D}_\mathcal{T}$. At each minibatch update, if an instance belongs to $\boldsymbol{D}_\mathcal{S}$ then the original loss function of Faster R-CNN is used for parameter update. If an instance belongs to $\boldsymbol{D}_\mathcal{T}$ then $q(y_c)$ in Eq. 4 and $q(\boldsymbol{y}_l)$ in Eq. 6 are used to refine the proposed bounding box annotations. $q(y_c)$ is used as the soft target labels in the cross entropy loss function for the mis-classification term and $(\bar{\boldsymbol{y}}_l+\alpha\tilde{\boldsymbol{y}}_l)/(\alpha+1)$ is used as the target location for the regression term. The modifications are made only in the ROI classifier loss function because the RPN is class agnostic.

False Negative Correction: Thus far, the robust detection method only refines the object proposals generated in phase one. This allows the model to correct false positive detections, *i.e.*, instances that do not contain any foreground object or that contain an object from a class different than the predicted class. However, we would also like to correct false negative predictions, *i.e.*, positive instances of foreground classes that are not detected in phase one.

To correct false negative instances, we rely on the hard negative mining phase of Faster R-CNN. In this phase a set of hard negative instances are added as background instances to the training set. Hard negatives that come from $\boldsymbol{D}_{\mathcal{S}}$ are actually background images. However, the "background" instances that are extracted from $\boldsymbol{D}_{\mathcal{T}}$ may be false negatives of phase one and may contain foreground objects. Therefore, during training for negative samples that belong to $\boldsymbol{D}_{\mathcal{T}}$, we define $p_{img}(y_c)$ to be a softened one-hot vector by setting the probability of a background to $1-\epsilon$ and the probability of the other class labels uniformly to ϵ/C. This is used as a soft target label in the cross-entropy loss.

3.4. Image Classification:

Phase two of our framework uses an image classification model to re-score bounding box proposals obtained in phase one. The image classification network is trained in a semi-supervised setting on top of images cropped from both $\boldsymbol{D}_{\mathcal{S}}$ (clean training set) and $\boldsymbol{D}_{\mathcal{T}}$ (noisy labeled set). For images in $\boldsymbol{D}_{\mathcal{S}}$, we use the cross-entropy loss against ground truth labels, but, for images in $\boldsymbol{D}_{\mathcal{T}}$ the cross-entropy loss is computed against soft labels obtained by Eq 2, where the weighted geometric mean between predicted classification score and a softened one-hot annotation vector is computed. This corresponds to multiclass extension of [56] which allows the classification model to refine noisy class labels for images in $\boldsymbol{D}_{\mathcal{T}}$.

Note that both $\boldsymbol{D}_{\mathcal{S}}$ and $\boldsymbol{D}_{\mathcal{T}}$ have bounding boxes annotations from foreground classes (although instances in $\boldsymbol{D}_{\mathcal{T}}$ have noisy labels). For training the image classification models, we augment these two datasets with bounding boxes mined from areas in the image that do not have overlap with bounding boxes in $\boldsymbol{D}_{\mathcal{S}}$ or $\boldsymbol{D}_{\mathcal{T}}$.

4. Experiments

To compare with state-of-the-art methods we follow the experimental design of [2]. We perform three experiments on three source/target domains and use similar hyper-parameters as [2]. We use the Faster R-CNN implementation available in the object detection API [25] source code. In all the experiments, including the baselines and our method, we set the initial learning rate to 0.001 for $50,000$ iterations and reduce it to 0.0001 for the next $20,000$ iterations (a similar training scheme as [2]). We linearly anneal α from 100 to 0.5 for the first $50,000$ iterations and keep it constant thereafter. We use InceptionV2 [52], pre-trained on ImageNet [8], as the backbone for Faster R-CNN. In one slight departure, we set aside a small portion of the training set as validation for setting hyper-parameters. InceptionV4 [51] is used for the image classification phase with initial learning rate of 3×10^{-4} that drops every 2 epochs by a factor 0.94. We set the batch size to 32 and train for $300\,000$ steps.

Baselines: We compare our method against the following progressively more sophisticated baselines.

- **Faster R-CNN** [47]: This is the most primitive baseline. A Faster R-CNN object detector is trained on the source domain and tested on the target domain so that the object detector is blind to the target domain.
- **Pseudo-labeling** [26]: A simplified version of our method in which Faster R-CNN is trained on the source domain to extract object proposals in the target domain, and then based on a pre-determined threshold, a subset of the object proposals are selected and used for fine-tuning Faster R-CNN. This process can be repeated. This method corresponds to the special case where $\alpha = 0$ is fixed throughout training. The original method in [26] performs a progressive adaptation, which is computationally extensive. Since our method and the previous state-of-the-art method perform only one extra fine-tuning step, we perform only one repetition for a fair comparison.
- **Feature Learning** [2]: This state-of-the-art domain adaptation method reduces the domain discrepancy by learning robust features in an adversarial manner. We follow the experimental setup used in [2].

Datasets: Following [2] we evaluate performance on multi- and single-label object detection tasks using three different datasets. Depending on the experiment, some datasets are used as both target and source domains and some are only used as either the source or target domain.

- ***SIM 10K*** [28] is a simulated dataset containing $10,000$ images synthesized by the Grand Theft Auto game engine. In this dataset, which simulates car driving scenes captured by a dash-cam, there are $58,701$ annotated car instances with bounding boxes. We use 10% of these for validation and the remainder for training.
- ***Cityscapes*** [4] is a dataset[4] of real urban scenes containing $3,475$ images captured by a dash-cam, $2,975$ images are used for training and the remaining 500 for validation. Following [2] we report results on the validation set because the test set doesn't have annotations. In our experiments we used the tightest bounding box of an instance segmentation mask as ground truth. There are 8 different object categories in this dataset including *person, rider, car, truck, bus, train, motorcycle and bicycle*.
- ***Foggy Cityscapes*** [48] is the foggy version of *Cityscapes*. The depth maps provided in *Cityscapes* are used to simulate three intensity levels of fog in [48]. In our experiments we used the fog level with

[4]This dataset is usually used for instance segmentation and not object detection.

Cityscapes → Foggy Cityscapes													
Method	Cls-Cor	Box-R	FN-Cor	person	rider	car	truck	bus	train	motorcycle	bicycle	mAP
Faster R-CNN[47]				31.69	39.41	45.81	23.86	39.34	20.64	22.26	32.36	31.92
Pseudo-labeling[26]				31.94	39.94	47.97	25.13	39.85	27.22	25.01	34.12	33.90
Feature Learning [2]				**35.81**	41.63	47.36	28.49	32.41	**31.18**	26.53	34.26	34.70
Noisy Labeling (Ours) :	✓	✗	✗	34.82	41.89	48.93	27.68	42.53	26.72	26.65	35.76	35.62
	✓	✓	✗	35.26	**42.86**	**50.29**	27.87	42.98	25.43	25.30	35.94	36.06
	✓	✓	✓	35.10	42.15	49.17	**30.07**	**45.25**	26.97	**26.85**	**36.03**	**36.45**
Faster R-CNN[47] trained on target				40.63	47.05	62.50	33.12	50.43	39.44	32.57	42.43	43.52

Table 2: Quantitative results comparing our method to baselines for adapting from *Cityscapes* to *Foggy Cityscapes*. We record the average precision (AP) on the *Cityscapes* validation set. "Cls-Cor" represents "classification error correction", Box-R stands for "Bounding Box Refinement" component, and FN-Cor stands for "False Negative Correction" component of our method. The last row shows the base detector's performance if labeled data for target domain was available.

highest intensity (least visibility). The same dataset split used for *Cityscapes* is used for *Foggy Cityscapes*.

- ***KITTI*** [16] is another real-world dataset consisting of $7,481$ images of real-world traffic situations, including freeways, urban and rural areas. Following [2] we used the whole dataset for both training, when it is used as source, and test, when it is used as target.

4.1. Adapting synthetic data to real world

In this experiment, the detector is trained on synthetic data generated using computer simulations and the model is adapted to real world examples. This is an important use case as it circumvents the lack of annotated training data common to many applications (*e.g.* autonomous driving). The source domain is *SIM 10K* and the target domain is *Cityscapes* dataset (denoted by "*SIM 10K* → *Cityscapes*"). We use the validation set of *Cityscapes* for evaluating the results. We only train the detector on annotated *cars* because *cars* is the only object common to both *SIM 10K* and *Cityscapes*.

SIM 10K → Cityscapes				
Method	Cls-Cor	Box-R	FN-Cor	AP
Faster R-CNN[47]				31.08
Pseudo-labeling[26]				39.05
Feature Learning [2]				40.10
Noisy Labeling (Ours) :	✓	✗	✗	**41.28**
	✓	✓	✗	**41.83**
	✓	✓	✓	**42.56**
Faster R-CNN[47] trained on target				68.10

Table 1: Quantitative results comparing our method to baselines for adapting from *SIM 10K* dataset to *Cityscapes*. We record average precision (AP) on the *Cityscapes* validation set. The last row shows the base detector's performance if labeled data for target domain was available.

Table 1 compares our method to the baselines. We tested our method with "Classification Error Correction (Cls-Cor)"[5], with or without the "Bounding Box Refinement (Box-R)" and "False Negative Correction (FN-Cor)" components. The state-of-the-art Feature Learning [2] method has $+1.05\%$ improvement over the basic Pseudo-labeling[26] baseline. Our best performing method has a $+3.51\%$ improvement over the same baseline yielding more than triple the improvement over the incumbent state-of-the-art.

4.2. Adapting normal to foggy weather

Changes in weather conditions can significantly affect visual data. In applications such as autonomous driving, the object detector must perform accurately in all conditions [48]. However, it is often not possible to capture all possible variations of objects in all weather conditions. Therefore, models must be adaptable to differing weather conditions. Here we evaluate our method and demonstrate its superiority over the current state-of-the-art for this task. We use *Cityscapes* dataset as the source domain and *Foggy Cityscapes* as the target domain (denoted by "*Cityscapes* → *Foggy Cityscapes*").

Table. 2 compares our method to the baselines on multi-label domain adaptation. The categories in this experiment are *person, rider, car, truck, bus, train, motorcycle, bicycle*. Average precision for each category along with the mean average precision (mAP) of all the objects are reported. Our method improves Faster R-CNN mAP by $+4.53\%$, while the state-of-the-art's improvement is $+2.78\%$.

4.3. Adapting to a new dataset

The previous examples of domain adaptation (synthetic data and weather change) are somewhat specialized. How-

[5]Turning off Cls-Cor reduces our approach to a method similar to Pseudo-labeling[26] with similar performance. To maintain robustness to label noise, we run all experiments with Cls-Cor component.

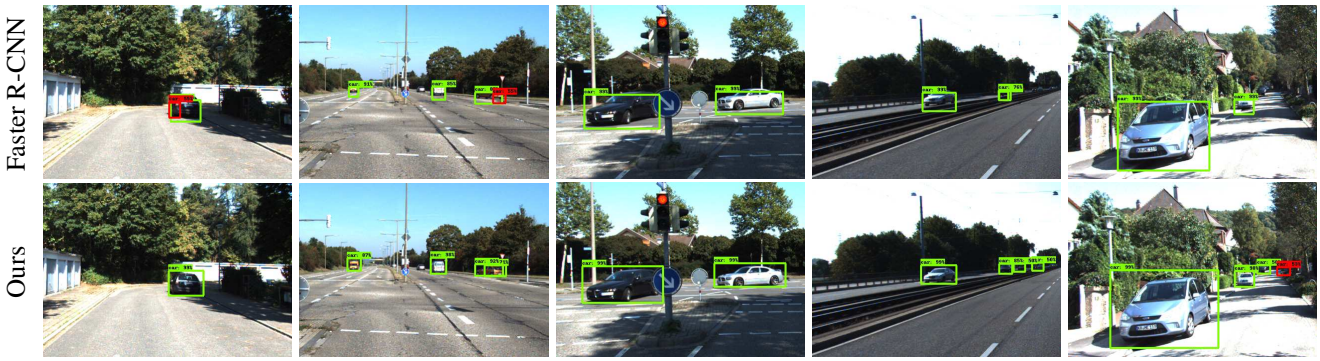

Figure 2: Qualitative comparison of our method with Faster R-CNN on the "*Cityscapes* → *KITTI*" experiment. Each column corresponds to a particular image in the *KITTI* test set. Top and bottom images in each column illustrate the bounding boxes of the cars detected by Faster R-CNN and our method respectively. In the first two columns our method corrects several false positives. In all cases our method successfully corrected the size/location of the bounding boxes (*e.g.* the rooflines in the third column). In the fourth and fifth examples, our method has detected cars that Faster R-CNN has missed. Nevertheless, false positives do occur (*e.g.* in column five), though the probability of those specific false positives is low (53% in this example).

ever, any change in camera (*e.g.* angle, resolution, quality, type, etc.) or environmental setup can cause domain shift. We investigate the ability of our method to adapt from one real dataset to another real dataset. We use *Cityscapes* and *KITTI* as the source and target domain in two separate evaluations. We denote the experiment in which *Cityscapes* is the source domain and *KITTI* is the target domain by "*Cityscapes* → *KITTI*", and vice versa by "*KITTI* → *Cityscapes*".

Tables 3 and 4 compare average precision on the *car* class, the only common object. Our method significantly outperforms the state-of-the-art in both situations (*Cityscapes* ⇄ *KITTI*). Qualitative results of our method on the *KITTI* test set are shown in Figure 2.

5. Conclusion

Domain shift can severely limit the real-world deployment of object-detection-based applications when labeled data collection is either expensive or infeasible. We have proposed an unsupervised approach to mitigate this problem by formulating the problem as robust learning. Our robust object detection framework copes with labeling noise on both object classes and bounding boxes. State-of-the-art performance is achieved by robust training in the target domain using a model trained only in the source domain. This approach eliminates the need for collecting data in the target domain and integrates other sources of information using detection re-scoring.

6. Acknowledgements

This work was supported by the Security R&D Group of SK Telecom.

KITTI → *Cityscapes*				
Method	Cls-Cor	Box-R	FN-Cor	AP
Faster R-CNN[47]				31.10
Pseudo-labeling[26]				40.23
Feature Learning [2]				40.57
Noisy Labeling (Ours) :	✓	✗	✗	**42.03**
	✓	✓	✗	**42.39**
	✓	✓	✓	**42.98**
Faster R-CNN[47] trained on target				68.10

Table 3: Quantitative comparison of our method with baselines for adapting from *KITTI* to *Cityscapes*. We record average precision (AP) on the *Cityscapes* test set. The last row gives the base detector's performance if labeled data for the target domain was available.

Cityscapes → *KITTI*				
Method	Cls-Cor	Box-R	FN-Cor	AP
Faster R-CNN[47]				56.21
Pseudo-labeling[26]				73.84
Feature Learning [2]				73.76
Noisy Labeling (Ours) :	✓	✗	✗	**76.36**
	✓	✓	✗	**76.93**
	✓	✓	✓	**77.61**
Faster R-CNN[47] trained on target				90.13

Table 4: Quantitative comparison of our method with baselines for adapting *Cityscapes* to *KITTI*. We record average precision (AP) on the *KITTI* train set. The last row gives the base detector's performance if labeled data for target domain was available.

References

[1] Pau Panareda Busto and Juergen Gall. Open set domain adaptation. In *ICCV*, pages 754–763, 2017. 2

[2] Yuhua Chen, Wen Li, Christos Sakaridis, Dengxin Dai, and Luc 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. 1, 2, 6, 7, 8

[3] Yuhua Chen, Wen Li, and Luc Van Gool. Road: Reality oriented adaptation for semantic segmentation of urban scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7892–7901, 2018. 2

[4] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016. 6

[5] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-fcn: Object detection via region-based fully convolutional networks. In *Advances in neural information processing systems*, pages 379–387, 2016. 2

[6] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005. 2

[7] Mostafa Dehghani, Arash Mehrjou, Stephan Gouws, Jaap Kamps, and Bernhard Schölkopf. Fidelity-weighted learning. In *International Conference on Learning Representations (ICLR)*, 2018. 2

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 6

[9] Lixin Duan, Ivor W Tsang, and Dong Xu. Domain transfer multiple kernel learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(3):465–479, 2012. 2

[10] Lixin Duan, Dong Xu, Ivor Wai-Hung Tsang, and Jiebo Luo. Visual event recognition in videos by learning from web data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(9):1667–1680, 2012. 2

[11] 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. 2

[12] Basura Fernando, Amaury Habrard, Marc Sebban, and Tinne Tuytelaars. Unsupervised visual domain adaptation using subspace alignment. In *Proceedings of the IEEE international conference on computer vision*, pages 2960–2967, 2013. 2

[13] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. *arXiv preprint arXiv:1409.7495*, 2014. 2

[14] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030, 2016. 1, 2

[15] Timnit Gebru, Judy Hoffman, and Li Fei-Fei. Fine-grained recognition in the wild: A multi-task domain adaptation approach. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 1358–1367. IEEE, 2017. 2

[16] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013. 7

[17] Muhammad Ghifary, W Bastiaan Kleijn, Mengjie Zhang, David Balduzzi, and Wen Li. Deep reconstruction-classification networks for unsupervised domain adaptation. In *European Conference on Computer Vision*, pages 597–613. Springer, 2016. 2

[18] Ross Girshick. Fast r-cnn. pages 1440–1448, 2015. 2

[19] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014. 2

[20] Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2066–2073. IEEE, 2012. 2

[21] Raghuraman Gopalan, Ruonan Li, and Rama Chellappa. Domain adaptation for object recognition: An unsupervised approach. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 999–1006. IEEE, 2011. 2

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *European conference on computer vision*, pages 346–361. Springer, 2014. 2

[23] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei A Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. *arXiv preprint arXiv:1711.03213*, 2017. 1

[24] Judy Hoffman, Dequan Wang, Fisher Yu, and Trevor Darrell. Fcns in the wild: Pixel-level adversarial and constraint-based adaptation. *arXiv preprint arXiv:1612.02649*, 2016. 2

[25] Jonathan Huang, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Ian Fischer, Zbigniew Wojna, Yang Song, Sergio Guadarrama, et al. Speed/accuracy trade-offs for modern convolutional object detectors. In *IEEE CVPR*, volume 4, 2017. 2, 6

[26] Naoto Inoue, Ryosuke Furuta, Toshihiko Yamasaki, and Kiyoharu 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. 6, 7, 8

[27] Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. Mentornet: Regularizing very deep neural networks on corrupted labels. In *International Conference on Machine Learning (ICML)*, 2018. 2

[28] Matthew Johnson-Roberson, Charles Barto, Rounak Mehta, Sharath Nittur Sridhar, Karl Rosaen, and Ram Vasudevan.

Driving in the matrix: Can virtual worlds replace human-generated annotations for real world tasks? *arXiv preprint arXiv:1610.01983*, 2016. 6

[29] Mehran Khodabandeh, Hamid Reza Vaezi Joze, Ilya Zharkov, and Vivek Pradeep. Diy human action dataset generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1448–1458, 2018. 2

[30] Brian Kulis, Kate Saenko, and Trevor Darrell. What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1785–1792. IEEE, 2011. 2

[31] Wen Li, Zheng Xu, Dong Xu, Dengxin Dai, and Luc Van Gool. Domain generalization and adaptation using low rank exemplar svms. *IEEE transactions on pattern analysis and machine intelligence*, 40(5):1114–1127, 2018. 2

[32] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016. 2

[33] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I Jordan. Learning transferable features with deep adaptation networks. *arXiv preprint arXiv:1502.02791*, 2015. 2

[34] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Unsupervised domain adaptation with residual transfer networks. In *Advances in Neural Information Processing Systems*, pages 136–144, 2016. 2

[35] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Deep transfer learning with joint adaptation networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2208–2217. JMLR. org, 2017. 1

[36] Arash Mehrjou, Mehran Khodabandeh, and Greg Mori. Distribution aware active learning. *arXiv preprint arXiv:1805.08916*, 2018. 2

[37] Ishan Misra, C. Lawrence Zitnick, Margaret Mitchell, and Ross Girshick. Seeing through the human reporting bias: Visual classifiers from noisy human-centric labels. In *CVPR*, 2016. 2

[38] Volodymyr Mnih and Geoffrey E. Hinton. Learning to label aerial images from noisy data. In *International Conference on Machine Learning (ICML)*, pages 567–574, 2012. 2

[39] Saeid Motiian, Marco Piccirilli, Donald A Adjeroh, and Gianfranco Doretto. Unified deep supervised domain adaptation and generalization. In *The IEEE International Conference on Computer Vision (ICCV)*, volume 2, page 3, 2017. 2

[40] Zak Murez, Soheil Kolouri, David Kriegman, Ravi Ramamoorthi, and Kyungnam Kim. Image to image translation for domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4500–4509, 2018. 1

[41] Nagarajan Natarajan, Inderjit S. Dhillon, Pradeep K. Ravikumar, and Ambuj Tewari. Learning with noisy labels. In *Advances in neural information processing systems*, pages 1196–1204, 2013. 2

[42] Giorgio Patrini, Alessandro Rozza, Aditya Menon, Richard Nock, and Lizhen Qu. Making neural networks robust to label noise: A loss correction approach. In *Computer Vision and Pattern Recognition*, 2017. 2

[43] Anant Raj, Vinay P Namboodiri, and Tinne Tuytelaars. Subspace alignment based domain adaptation for rcnn detector. *arXiv preprint arXiv:1507.05578*, 2015. 2

[44] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. 2

[45] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. *arXiv preprint*, 2017. 2

[46] Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. In *International Conference on Machine Learning (ICML)*, 2018. 2

[47] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 1, 2, 3, 6, 7, 8

[48] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Semantic foggy scene understanding with synthetic data. *International Journal of Computer Vision*, pages 1–20, 2018. 6, 7

[49] Sainbayar Sukhbaatar, Joan Bruna, Manohar Paluri, Lubomir Bourdev, and Rob Fergus. Training convolutional networks with noisy labels. *arXiv preprint arXiv:1406.2080*, 2014. 2

[50] Baochen Sun, Jiashi Feng, and Kate Saenko. Return of frustratingly easy domain adaptation. In *AAAI*, volume 6, page 8, 2016. 2

[51] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017. 6

[52] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016. 6

[53] Daiki Tanaka, Daiki Ikami, Toshihiko Yamasaki, and Kiyoharu Aizawa. Joint optimization framework for learning with noisy labels. In *Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[54] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7167–7176, 2017. 1

[55] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2):154–171, 2013. 2

[56] Arash Vahdat. Toward robustness against label noise in training deep discriminative neural networks. In *Neural Information Processing Systems (NIPS)*, 2017. 2, 4, 6

[57] Arash Vahdat and Greg Mori. Handling uncertain tags in visual recognition. In *International Conference on Computer Vision (ICCV)*, 2013. 2

[58] Arash Vahdat, Guang-Tong Zhou, and Greg Mori. Discovering video clusters from visual features and noisy tags. In *European Conference on Computer Vision (ECCV)*, 2014. 2

[59] Andreas Veit, Neil Alldrin, Gal Chechik, Ivan Krasin, Abhinav Gupta, and Serge Belongie. Learning from noisy large-scale datasets with minimal supervision. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6575–6583. IEEE, 2017. 2

[60] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. 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

[61] Tuan-Hung Vu, Himalaya Jain, Maxime Bucher, Matthieu Cord, and Patrick Pérez. Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2517–2526, 2019. 2

[62] Yifei Wang, Wen Li, Dengxin Dai, and Luc Van Gool. Deep domain adaptation by geodesic distance minimization. *arXiv preprint arXiv:1707.09842*, 2017. 2

[63] Tong Xiao, Tian Xia, Yi Yang, Chang Huang, and Xiaogang Wang. Learning from massive noisy labeled data for image classification. In *Computer Vision and Pattern Recognition (CVPR)*, 2015. 2

[64] Jiaolong Xu, Sebastian Ramos, David Vázquez, and Antonio M López. Domain adaptation of deformable part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 36(12):2367–2380, 2014. 2

[65] Xiyu Yu, Tongliang Liu, Mingming Gong, and Dacheng Tao. Learning with biased complementary labels. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 68–83, 2018. 2

[66] Liliang Zhang, Liang Lin, Xiaodan Liang, and Kaiming He. Is faster r-cnn doing well for pedestrian detection? In *European Conference on Computer Vision*, pages 443–457. Springer, 2016. 2

[67] Yang Zhang, Philip David, and Boqing Gong. Curriculum domain adaptation for semantic segmentation of urban scenes. In *The IEEE International Conference on Computer Vision (ICCV)*, volume 2, page 6, 2017. 2

[68] Zhilu Zhang and Mert R Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. In *Neural Information Processing Systems (NIPS)*, 2018. 2

[69] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2223–2232, 2017. 1