MVUDA: Unsupervised Domain Adaptation for Multi-view Pedestrian Detection

Citation for the original published paper (version of record):
Brorsson, E., Svensson, L., Bengtsson, K. et al (2026). MVUDA: Unsupervised Domain Adaptation for Multi-view Pedestrian Detection. Machine Vision and Applications, 37(1).
http://dx.doi.org/10.1007/s00138-025-01764-y

(article starts on next page)

Machine Vision and Applications (2026) 37:6
https://doi.org/10.1007/s00138-025-01764-y

RESEARCH

MVUDA: Unsupervised Domain Adaptation for Multi-view Pedestrian Detection

Erik Brorsson[1,2] · Lennart Svensson[2] · Kristofer Bengtsson[1] · Knut Åkesson[2]

Received: 3 June 2025 / Revised: 14 September 2025 / Accepted: 19 October 2025

Abstract

We address multi-view pedestrian detection in a setting where labeled data is collected using a multi-camera setup different from the one used for testing. While recent multi-view pedestrian detectors perform well on the camera rig used for training, their performance declines when applied to a different setup. To facilitate seamless deployment across varied camera rigs, we propose an unsupervised domain adaptation (UDA) method that adapts the model to new rigs without requiring additional labeled data. Specifically, we leverage the mean teacher self-training framework with a novel pseudo-labeling technique tailored to multi-view pedestrian detection. This method achieves state-of-the-art performance on multiple benchmarks, including MultiviewX→Wildtrack. Unlike previous methods, our approach eliminates the need for external labeled monocular datasets, thereby reducing reliance on labeled data. Extensive evaluations demonstrate the effectiveness of our method and validate key design choices. By enabling robust adaptation across camera setups, our work enhances the practicality of multi-view pedestrian detectors and establishes a strong UDA baseline for future research.

Keywords Multi-view object detection · Unsupervised domain adaptation · Self-training · Pseudo-labeling

Introduction

Multi-view detection aims to detect objects from a set of images captured simultaneously by multiple cameras, each providing a distinct view of the same scene. Using multiple views allows for greater robustness to occlusions and facilitates inferring 3D properties of objects, which can be challenging with a single camera. In this paper, we focus on multi-view pedestrian detection, where the goal is to detect pedestrians and estimate their location on a ground plane using images captured by multiple stationary cameras. This task is relevant in applications like surveillance [1], robotics [2], sports analytics [3], and autonomous mobile robot control [4].

Recent methods for multi-view pedestrian detection consider all input images jointly to learn a dense feature map in bird's-eye-view (BEV) [5–10]. This BEV representation, which is aligned with the ground plane, is then refined, typically with convolutional layers, to obtain an occupancy map describing likely pedestrian locations. Finally, the occupancy map is post-processed via thresholding and non-maximum suppression to derive pedestrian detections. Although these methods have achieved impressive results, they rely on labeled multi-view datasets, which are typically scarce due to the costs of multi-camera setups and image annotation. In practice, labeled data is typically limited to simulations or a single real-world camera rig, leading to overfitting and poor generalization across different camera setups.

Collecting unlabeled data from the real-world test setup, however, is relatively straight-forward, making unsupervised domain adaptation (UDA) a promising solution to the generalization challenges in multi-view detection. UDA is well established for monocular perception tasks such as image classification, semantic segmentation, and object detection, with mean teacher self-training as a popular approach [11–13]. This approach trains a student model on

✉ Erik Brorsson
erik.brorsson@volvo.com

Lennart Svensson
lennart.svensson@chalmers.se

Kristofer Bengtsson
kristofer.bengtsson@volvo.com

Knut Åkesson
knut.akesson@chalmers.se

[1] Global Trucks Operations, Volvo Group, Göteborg, Sweden

[2] Department of Electrical Engineering, Chalmers University of Technology, Göteborg, Sweden

Published online: 21 November 2025

unlabeled data using pseudo-labels generated by a mean teacher [14], an exponential moving average of the student's parameters. However, to the best of our knowledge, Lima et al. [15, 16] constitute the only works to explore UDA in multi-view pedestrian detection. In their approach, they adapt a multi-view detector through self-training, but rely on a pre-trained external detector based on large, labeled monocular datasets, limiting practicality for applications without access to such resources.

We address this gap by considering a strict UDA setting that excludes any external labeled dataset or pre-trained detector. Apart from its practical relevancy due to restrictive licensing of datasets and derived detectors, it is also conceptually interesting as it opens possibilities to extend the framework to new object types in the future. We build on mean teacher self-training, adapting it for multi-view pedestrian detection and identifying key success factors for the strict UDA settings. Importantly, we propose a novel pseudo-labeling method to enhance pseudo-label reliability, significantly improving self-training efficacy. Our method achieves state-of-the-art performance across multiple benchmarks. Furthermore, while recent works primarily focus on bridging simulated and real-world domains, few consider the challenges posed by changing camera configurations. To facilitate this, we introduce two new benchmarks specifically for cross-camera rig adaptation.

Our contributions can be summarized as follows:

1. We unveil the potential of self-training for multi-view pedestrian detection under a strict UDA setting and develop a state-of-the-art method for this problem.
2. We propose a simple yet effective pseudo-labeling method that improves pseudo-label reliability and thereby the effectiveness of self-training.
3. We demonstrate the efficacy of our method on multiple established benchmarks and on two new benchmarks, which we introduce to specifically address cross-camera rig adaptation.

1 Related work

1.1 Multi-view pedestrian detection

Multi-view pedestrian detection aims to utilize cameras with different viewpoints to enable more robust detection and localization in 3D than what is possible with a single camera. Early methods relied on background subtraction in each view and inferred 3D ground plane positions using graphical models combined with Bayesian inference [17–19]. Since background subtraction is not sufficiently discriminative in crowded scenes, many later works replaced this component with more advanced methods of monocular perception, such as 2D bounding box detection [20–22], human pose estimation [20], or instance segmentation [23]. These methods also proposed alternative ways to fuse individual detections, such as projecting detections onto a ground plane and grouping them based on Euclidean proximity [20, 21, 23], or employing Conditional Random Fields (CRF) [22]. However, because these methods rely on monocular perception, any deficiencies in the individual views can degrade overall performance.

In contrast, end-to-end methods consider all input images jointly, enabling a more comprehensive understanding of correspondences across views. Early methods processed each view with a Convolutional Neural Network (CNN) to extract features and then applied either a Multilayer Perceptron (MLP) [24] or CRF [25] to generate detections by jointly considering these features. Recently, MVDet [5] introduced a new approach by projecting features from individual views into a bird's-eye view (BEV) through a perspective transformation, creating dense feature maps in BEV. Many recent methods build on this idea through improved perspective view feature extraction [26], enhanced feature aggregation in BEV [6, 7, 10], modified decoders [27, 28], and multi-view-specific data augmentation techniques [8, 9]. While these approaches continue to push the state-of-the-art in multi-view pedestrian detection, they require labeled multi-view datasets for training and typically fail to generalize well to new camera setups. In this work, we aim to relax the dependency on labeled multi-view data, making these methods more useful in practice.

1.2 Unsupervised domain adaptation (UDA)

Given a labeled dataset from a source domain and an unlabeled dataset from a target domain, Unsupervised Domain Adaptation (UDA) aims to transfer knowledge from the source to the target, allowing models to generalize to new data distributions without additional labels. UDA has been widely applied in computer vision tasks, including image classification [29–31], semantic segmentation [32–35], and object detection [11, 13, 36, 37]. Recent UDA methods largely follow two approaches: adversarial learning and self-training. Adversarial learning seeks to create domain-invariant input [11, 33, 34], output [31, 35] or features [13, 29, 32], helping the model to disregard variations across the domains that are irrelevant to the task. Self-training, on the other hand, involves training a student model in a supervised fashion on the target dataset using pseudo-labels [38]. To improve the quality of the pseudo-labels, many approaches [11–13, 36, 37] use a mean teacher [14], which is an exponential moving average of the student's parameters, to generate these labels during training. Nevertheless, incorrect pseudo-labels remain a significant challenge [13, 37, 39]. Furthermore, while UDA has shown substantial progress in monocular tasks, adapting it to multi-view perception remains largely unexplored.

In one of the few efforts to apply UDA methods to multi-view pedestrian detection, Lima et al. [15] proposed adapting the detector from [6] to unlabeled target data using self-training. However, the method suffered from low-quality pseudo-labels, resulting in modest improvements on a single benchmark. Lima et al. later improved their approached by incorporating a mean teacher for pseudo-labeling [16]. However, the success of the method is conditioned on pre-training with pseudo-labels generated by an external detector [20], which in turn relies on supervised training on large, labeled datasets for monocular human pose estimation. As a result, the approach still requires substantial amounts of labeled data, which may limit its practical use. In contrast to these methods, our work presents a solution for unsupervised domain adaptation in multi-view pedestrian detection that does not depend on any auxiliary labeled datasets or pre-trained models derived from them.

2 Methods

In this section, we introduce our UDA method for multi-view pedestrian detection, designed to leverage labeled source data alongside unlabeled target data to train a multi-view detector for deployment on the target domain. We begin by detailing the detector architecture. Thereafter, we outline our overall UDA strategy and, finally, introduce our approach for generating high-quality pseudo-labels.

2.1 Multi-view detector

Due to its simplicity and good generalization capability, we use the multi-view detector GMVD [6]. Like many other works [5–10], this detector produces an occupancy map that describes likely pedestrian locations on the ground plane. To derive the final prediction, which is a set of pedestrian locations on the ground plane, this occupancy map is post-processed via thresholding and non-maximum suppression. The detector consists of three components: 2D image feature extraction, perspective transformation, and spatial aggregation.

Feature extractor: Given N RGB-images from different views, a ResNet-18 [40] extracts features with C channels and spatial dimension $H_f \times W_f$ for each view.

Perspective transformation: Assuming known intrinsic and extrinsic calibration for each camera, the output of the feature extractor are transformed to BEV using a perspective transformation. The reader is referred to [5] for the full details on this transformation. The result of this operation is N BEV feature maps of shape $C \times H_g \times W_g$, where H_g and W_g defines the spatial dimension of the BEV. The purpose is to put all features in the common BEV, which prepares them for spatial aggregation. For a detailed explanation, we refer the reader to the original paper [5].

Spatial aggregation: The BEV features from different cameras are concatenated to produce a BEV feature map of shape $N \times C \times H_g \times W_g$. Average pooling is then applied along the first dimension to reduce its shape to $C \times H_g \times W_g$. Since average pooling makes the shape of the BEV feature map independent of the number of views N, it allows for naturally handling a varying number of cameras. Finally, three dilated convolutional layers process the BEV feature map to regress an occupancy map of dimension $H_g \times W_g$. During inference, the probabilistic occupancy map is thresholded to produce detection candidates, which are then subject to non-maximum suppression (NMS) to remove duplicate detections.

2.2 Mean teacher self-training

In multi-view detection, a labeled source dataset with N^s samples can be described as $\mathcal{D}^S = \{(x^{S,k}, y^{S,k})\}_{k=1}^{N^S}$, where $x^{S,k}$ denotes a batch of multi-view images from the source domain and $y^{S,k}$ denotes the associated label. The label $y^{S,k}$ is a matrix of shape $H_g \times W_g$ that describes the pedestrians' locations on the ground plane. Specifically, each pedestrian is associated with the closest cell in the matrix and assigns that cell with the value 1, while all other entries have the value 0. Similarly, an unlabeled target dataset with N^T samples is described by $\mathcal{D}^T = \{x^{T,k}\}_{k=1}^{N^T}$, where $x^{T,k}$ is a batch for multi-view images from the target domain.

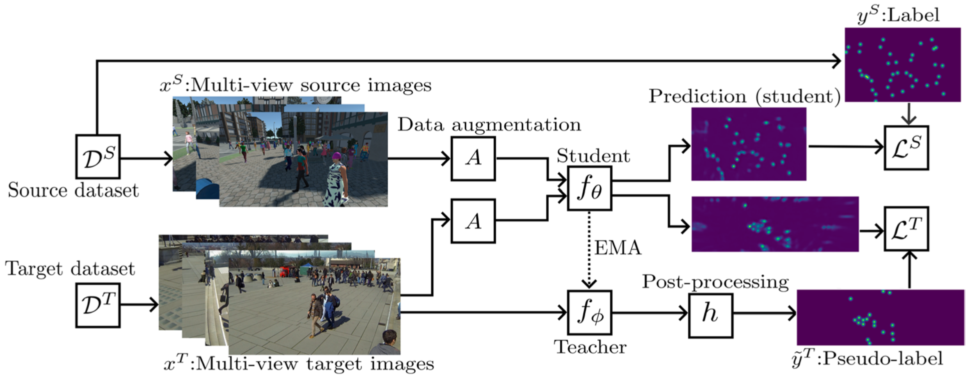


Fig. 1 An overview of our proposed self-training method for UDA multi-view pedestrian detection. A student is trained with labels on the source domain and pseudo-labels on the target domain, which are created by a mean teacher. While the teacher creates pseudo-labels on unaugmented data, the student receives strongly augmented images. Note that the label and pseudo-label have been *softened* with a Gaussian kernel in this figure to ease visualization

In established self-training methods for monocular perception, a model f_θ (the student) is trained on labeled samples from the source dataset and pseudo-labeled samples from the target dataset. Note that f_θ in our case is the multi-view detector described in the previous section. Moreover, the pseudo-labels are typically created during training by a mean teacher f_ϕ. The architecture of f_ϕ is the same as f_θ, but its weights ϕ are updated as an exponential moving average of the student's weights θ according to

$$\phi_{t+1} \leftarrow \alpha\phi_t + (1-\alpha)\theta_t, \tag{1}$$

where α is a hyperparameter. Formally, the pseudo-label $\tilde{y}^T$ for a batch of multi-view images x^T on the target domain (dropping the index *k* for ease of notation) is defined by

$$\tilde{y}^T = h(f_\phi(x^T)), \tag{2}$$

where h denotes the post-processing function that maps the predictions to pseudo-labels. In multi-view pedestrian detection, h typically consist of applying a threshold to the predicted occupancy map and then applying non-maximum suppression. In this work, we consider both conventional post-processing and our own proposal, which will be described in the next section. Furthermore, while f_ϕ is fed target images x^T for pseudo-labeling, the student is fed augmented images $A(x^T)$. In our work, the same data augmentation method is also applied to the source images x^S to further support the student's generalization capability. Thus, the weights θ of the student network f_θ are trained to minimize the loss

$$L(\theta) = \mathbb{E}[\mathcal{L}^S(y^S, f_\theta(A(x^S))) + \lambda\mathcal{L}^T(\tilde{y}^T, f_\theta(A(x^T)))], \tag{3}$$

where the expectation is taken over data from the source and target datasets and λ is a hyperparameter that adjusts the influence of the target data. Following [5], we apply a Gaussian kernel $G(\cdot)$ to generate a *soft* target and train the model with the MSE loss. We adopt this loss for both the source and target domain according to

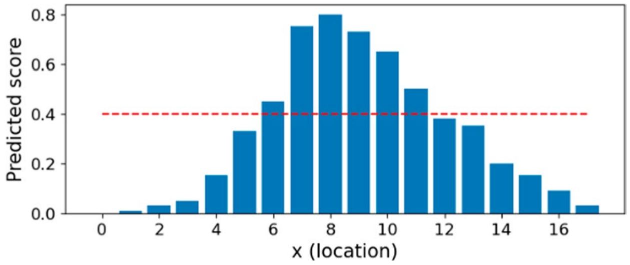


Fig. 2 Illustrative example of predicted occupancy scores in one dimension

$$\mathcal{L}^S(y,\hat{y}) = \mathcal{L}^T(y,\hat{y}) = \sum_{i=1}^{H_g}\sum_{j=1}^{W_g}(G(y_{ij}) - \hat{y}_{ij})^2, \tag{4}$$

where *y* and $\hat{y}$ denotes a label (or pseudo-label) and prediction respectively. The proposed mean teacher self-training framework is schematically illustrated in Fig. 1. Before adapting the model to the target domain, however, we pretrain it using only source data.

2.3 Local-max pseudo-labeling

An essential step in the self-training framework detailed in the previous section is the creation of pseudo-labels. In multi-view pedestrian detection, post-processing is applied to the predicted probabilistic occupancy map to derive a set of detections. In this section, we first review the conventional post-processing method and then introduce our alternative, which is tailored for the UDA problem.

Vanilla pseudo-labeling: The conventional method, adopted by e.g. [5, 6, 8, 28], comprises the following steps: First, all candidate locations with confidence scores exceeding a threshold τ are added to a list, sorted in descending order by score. Second, the algorithm selects the first candidate in the list as a detection and removes all candidates within a Euclidean distance d of this detection. Third, the second step is repeated until the list is empty.

To illustrate, consider a one-dimensional example with $\tau = 0.4$ and $d = 2$, shown in Fig. 2. Here, six candidates on positions $x \in \{6, 7, 8, 9, 10, 11\}$ exceed the threshold and are added to the list. Since position $x = 8$ has the highest confidence, it is selected as the first detection. Subsequently, candidates at positions 6,7,9, and 10 are removed from the list because they fall within distance d of the first detection. The candidate at position 11 is then selected as a second detection. The algorithm terminates at this point since no candidates remain in the list. Note, however, that if the threshold τ had been lower, a third detection at, e.g., $x = 5$ could have been attained.

Since the predicted confidence level on the target domain is difficult to foresee, we question whether this post-processing method is overly dependent on the threshold τ. Ideally, a well trained network is expected to exhibit predictions with a single local maximum at each location of a pedestrian following training with the MSE loss on the Gaussian targets described in Eq. (4). However, this post-processing method may also produce detections that are not local maxima. We hypothesize that such detections are less reliable, especially in UDA when the threshold τ is ambiguous.

Local-max pseudo-labeling: Motivated by the above analysis, we propose an alternative post-processing method that only considers points that are *local maxima* as candidate detections. To allow for an efficient implementation, in for example PyTorch, we define a local maxima as a position *ij* in the occupancy map for which the predicted confidence $\hat{y}_{ij}$ satisfies

$$\hat{y}_{ij} \geq \hat{y}_{kl} \, \forall k \in [i - k_d, i + k_d] \text{ and } \forall l \in [j - k_d, j + k_d], \tag{5}$$

where k and l are integers and k_d is a parameter that defines the size of the considered neighborhood. Since the predictions are expected to exhibit some degree of noise, we also require the predicted confidence of any detections to exceed the threshold τ. Note, however, that a location that is not a local maxima is never considered a candidate detection in our method, regardless of the value of τ, which distinguishes it from the conventional method. In the one-dimensional example illustrated in Fig. 2, the proposed post-processing method would not create a detection at $x = 11$, rather a single detection at $x = 8$ since this is the only local-maxima.

3 Experiments

3.1 Experimental setup

Datasets: we use the popular Wildtrack [41] and MultiviewX [5] datasets as well as a subset of the newly introduced GMVD [6] dataset. Wildtrack is a real-world dataset comprising 400 multi-view images collected from a single camera rig of seven cameras with overlapping fields of view, covering an area of 12×36 meters. For annotation, the ground plane is discretized into a 480×1440 grid, where each cell corresponds to a 2.5×2.5 cm region. Meanwhile, MultiviewX is a synthetic dataset of 400 images from six cameras covering an area of 16×25 meters, with a grid shape of 640×1000 of the same spatial resolution. GMVD is another synthetic dataset, distinct for its multiple scenes and camera configurations. Here, the covered area depends on the scene and the grid is chosen to attain the same spatial resolution of 2.5×2.5 cm.

To evaluate adaptation from labeled simulated data to unlabeled real-world data, and the converse, we consider the benchmarks MultiviewX→Wildtrack and Wildtrack→ MultiviewX. Following [6], we also consider the intra-dataset benchmarks Wildtrack 1,3,5,7→2,4,5,6, Wildtrack 2,4,5,6→1,3,5,7, and MultiviewX 1,2,6→3,4,5, where different subset of cameras from a single dataset constitute the source and target domain. The purpose is to evaluate adaptation across camera rigs without the presence of a sim-to-real domain gap. Additionally, to address this problem in the more challenging setting where the source and target datasets are collected from different scenes, we introduce two new benchmarks wherein GMVD and MultiviewX constitute the source and target domain respectively. Like the intra-dataset benchmarks, we consider labels on a single camera rig and therefore use only a subset of GMVD as the labeled source dataset. Specifically, we consider two different camera configurations on the first scene of GMVD as the source domain and introduce the benchmarks GMVD1→ MultiviewX and GMVD2→MultiviewX. For all benchmarks, we use the first 90% of samples in MultiviewX and Wildtrack for training and the last 10% for testing. GMVD1 and GMVD2 both consist of five cameras and comprise 517 training frames.

Evalation metrics: like most previous works, we evaluate the models in terms of the MODA, MODP, precision and recall metrics. MODA serves as the primary performance indicator, since it accounts for both missed detections and false positives, while MODP evaluates the localization precision [42]. For all metrics, we report the performance in percentage.

3.2 Implementation details

Following [6], input images are resized to shape 720x1280 before being processed by ResNet-18 [40], extracting 512-channel feature maps. These features are resized to shape 270x480 through bilinear interpolation before being projected to the ground plane, whose shape depends on the dataset. For spatial aggregation, we employ three convolutional layers with kernel size 3 and dilation factors of 1, 2 and 4. For training, we use the one-cycle learning rate scheduler [43] with a max learning rate of 0.1 and the SGD optimizer with momentum 0.5 and L2 regularization $5 \cdot 10^{-4}$. We use a batch size of 1 and employ early stopping to avoid overfitting. For evaluation, we use (conventional) NMS with a spatial threshold of 0.5 ms like previous works [5, 6]. However, while these works use the threshold $\tau = 0.4$, we evaluate the model on the range $\tau \in \{0.05, 0.10, ..., 0.95\}$ and select the result with highest MODA. The purpose is to ensure that the experimental results are not affected by the specific choice of τ, whose optimal value is unknown in the UDA setting.

Prior to self-training, we initialize ResNet-18 with ImageNet [44] weights and pre-train the model on only source data for 20 epochs, which constitutes our *Baseline*. Unless stated otherwise, the UDA results are obtained by adapting the baseline to the target domain by 5 epochs of self-training, using $\lambda = 1.0$, $\alpha = 0.99$, and the proposed local-max pseudo-labeling with $k_d = 3$. The threshold τ used for pseudo-labeling is experimentally set to 0.4 for MultiviewX→Wildtrack, 0.2 for Wildtrack→MultiviewX, and 0.3 for all other benchmarks. Moreover, Dropview [6] and 3DROM [8] augmentation is used both to train the baseline and in self-training.

In our experiments on MultiviewX→Wildtrack using an NVIDIA A100, training the baseline takes approximately five hours, and adaptation through five epochs of self-training requires an additional three hours. At inference, the GMVD model runs at about one second per frame in PyTorch on the same hardware. The main results presented in Section 4.3 are generated with the MatLab toolkit of MOTChallenge [45]. For convenience, all other results are generated with the Python implementation by [5], which was used during development of our method.

3.3 MVUDA compared with previous methods

In this section, we compare our adaptation method MVUDA with previous SOTA methods, as well as our *Baseline* (trained only on source), and the *Oracle*, which is trained

Table 1 Performance comparison with state-of-the-art methods on two cross-domain benchmarks. The methods marked with † rely on models trained on large, labeled datasets for monocular vision

Method	MODA	MODP	Precision	Recall
MultiviewX → Wildtrack				
†Lima et al. [16]	85.1	74.8	93.9	91.0
†PPM [23]	90.3	72.6	94.4	96.0
Oracle	91.3	75.5	97.0	94.2
GMVD [6]	70.7	73.8	89.1	80.6
TMVD [46]	74.9	**76.9**	90.4	83.8
MVFP [10]	82.6	76.2	89.6	**93.4**
Baseline	70.0	73.6	89.2	79.6
MVUDA (ours)	**85.4**	75.3	**96.5**	88.7
Wildtrack → MultiviewX				
†Lima et al. [16]	75.9	78.6	96.2	79.0
Oracle	90.8	82.2	97.3	93.4
GMVD* [6]	31.3	67.2	74.9	47.1
TMVD* [46]	60.0	74.4	88.2	69.3
Baseline	35.7	66.5	82.7	45.2
MVUDA (ours)	**82.2**	**75.6**	**93.2**	**88.7**

Table 2 Performance comparison with state-of-the-art methods on five different camera rig adaptation benchmarks

Method	MODA	MODP	Precision	Recall
Wildtrack 2,4,5,6 → 1,3,5,7				
Oracle	83.7	75.8	94.6	88.8
GMVD [6]	75.1	71.1	94.3	79.9
TMVD* [46]	78.7	73.2	95.2	82.9
Baseline	75.0	71.2	91.4	**82.8**
MVUDA (ours)	**79.2**	**78.0**	**96.2**	82.5
Wildtrack 1,3,5,7 → 2,4,5,6				
Oracle	86.9	71.6	94.3	92.4
GMVD [6]	62.6	67.4	86.7	73.9
TMVD* [46]	70.4	68.3	91.2	77.9
Baseline	71.6	68.5	87.8	83.2
MVUDA (ours)	**81.2**	**68.8**	**95.7**	**85.0**
MultiviewX 1,2,6→ 3,4,5				
Oracle	74.9	74.4	94.9	79.1
GMVD* [6]	46.1	68.9	85.5	55.5
TMVD* [46]	61.7	71.8	88.5	**71.0**
Baseline	54.0	69.4	89.3	61.4
MVUDA (ours)	**63.9**	**73.1**	**91.1**	70.8
GMVD1 → MultiviewX				
Oracle	90.8	82.2	97.3	93.4
GMVD* [6]	65.9	73.6	83.8	81.8
TMVD* [46]	75.9	76.6	91.0	84.2
Baseline	70.1	74.6	89.6	79.4
MVUDA (ours)	**88.9**	**78.4**	**97.0**	**91.8**
GMVD2 → MultiviewX				
Oracle	90.8	82.2	97.3	93.4
GMVD* [6]	61.5	73.0	91.1	68.1
TMVD* [46]	75.3	76.4	88.6	86.4
Baseline	66.3	74.3	85.5	79.9
MVUDA (ours)	**88.4**	**77.1**	**96.9**	**91.3**

with labels on the target domain similarly as the baseline was trained on the source domain. For qualitative results, please refer to the supplementary material. In Table 1, the results on MultiviewX→Wildtrack and Wildtrack→MultiviewX are presented. The dashed line separates the methods that use auxiliary labeled datasets from those that use labels only on the source domain. The bold figures indicate the highest scores in each column among the methods in the latter category. The results of previous methods are taken from the respective papers, although most only consider MultiviewX→Wildtrack. To extend the comparison further, we use the code of GMVD [6] and TMVD [46] to evaluate their performance on any remaining benchmarks as denoted by GMVD* and TMVD*. For GMVD*, we use the same training scheme as the authors. For TMVD*, we include pretraining on ImageNet along with Dropview and 3DROM data augmentation to enable a fair comparison with our baseline and MVUDA.

It can be seen that MVUDA boosts the baseline performance significantly with respect to all studied metrics on both benchmarks. MVUDA also achieves the highest MODA among the methods that don't rely on auxiliary labeled data. Impressively, our method boosts the baseline from 35.7 to 82.2 MODA on Wildtrack→MultiviewX, outperforming [16] by a large margin although they rely on a monocular detector derived from large, labeled monocular datasets.

In Table 2, we further evaluate MVUDA on five camera rig adaptation benchmarks. In all cases, our method significantly boosts the baseline in terms of MODA and outperforms [6] on all metrics. Furthermore, our method reaches close to Oracle performance on the two GMVD→MultiviewX benchmarks, which further highlights the effectiveness of our adaptation approach. Interestingly, the gap between MVUDA and the Oracle is slightly higher for the three intra-dataset benchmarks, suggesting that our method is less effective when the number of cameras is smaller.

It can also be seen that TMVD (and TMVD*) outperforms GMVD (and GMVD*) and the baseline by a large margin on most benchmarks. However, MVUDA consistently achieves the highest performance. While TMVD demonstrates strong generalizability across domains and camera setups, it also has notable drawbacks: its training time is roughly 75 h on an NVIDIA A100 compared to only five hours for our baseline (based on GMVD), and its inference time in PyTorch is around 9 s per frame versus about 1 s for GMVD, making it unsuitable for real-time use. In addition, TMVD relies on the assumption of known pedestrian dimensions. When this assumption does not hold, performance can degrade, and generalization to other object classes with greater intra-class variation is limited. For these reasons, we base our method MVUDA on the GMVD architecture.

3.4 Ablation study

To study the importance of Mean Teacher (MT) and data augmentation (Aug) in the self-training (ST) framework, we ablate these components on two benchmarks in Table 3.

Here, the first row shows the performance without any adaptation (baseline). Furthermore, self-training without mean teacher implies that the (frozen) baseline model creates pseudo-labels throughout training. It can be seen that self-training alone yields substantial improvements over the baseline. Moreover, the results improve significantly when adding the mean teacher and the data augmentation. It is noteworthy that the impact of data augmentation is greater on the sim-to-real benchmark, where it may serve as key component in overcoming the larger domain gap.

3.5 In-depth analysis of MVUDA

In this section, we analyze key components of our proposed method in detail, including the introduced pseudo-labeling technique, the parameter α, and the data augmentation. Unless stated otherwise, herein the UDA method comprises self-training using local-max pseudo-labeling with $k_d = 3$, $\alpha = 0.99$, $\lambda = 1$, and no data augmentation. Again, the threshold τ is set to 0.4 for MultiviewX→Wildtrack, 0.2 for Wildtrack→MultiviewX, and 0.3 for all other benchmarks, following the experiments presented in Table 4.

Table 3 Ablation of the Mean Teacher (MT) and data augmentation (Aug), which are two pivotal components in the self-training (ST) framework

ST	MT	Aug	MODA	MODP	Precision	Recall
MultiviewX → Wildtrack						
			70.0	73.6	89.2	79.6
✓			75.0	73.3	92.0	82.1
✓	✓		78.7	74.2	92.1	86.0
✓	✓	✓	**85.4**	**75.3**	**96.5**	**88.7**
GMVD1 → MultiviewX						
			70.3	74.5	89.7	79.5
✓			76.6	76.0	91.5	84.5
✓	✓		87.2	76.6	**97.6**	89.4
✓	✓	✓	**89.0**	**78.4**	97.0	**91.8**

Table 4 Performance comparison (MODA) of self-training with vanilla or local-max pseudo-labeling at different thresholds τ

Method	τ =0.2	0.25	0.3	0.35	0.4	0.45	0.5
MultiviewX ->Wildtrack (70.0)							
UDA vanilla	19.9	33.0	42.5	66.1	**78.6**	76.2	72.2
UDA local-max	55.8	67.1	70.8	74.4	75.8	73.9	66.7
WildTrack ->MultiViewX (35.9)							
UDA vanilla	0.0	0.0	48.1	47.6	47.9	30.5	20.1
UDA local-max	73.2	**75.7**	68.7	59.2	43.5	27.4	13.5
WildTrack 2,4,5,6 ->1,3,5,7 (75.2)							
UDA vanilla	0.0	24.1	73.8	77.4	78.5	78.0	75.2
UDA local-max	65.3	72.6	**78.6**	78.5	77.7	78.2	73.4
wildtrack 1,3,5,7 ->2,4,5,6 (72.3)							
UDA vanilla	0.4	24.9	57.9	75.3	73.4	53.4	37.7
UDA local-max	71.0	75.6	**79.8**	77.5	60.6	44.5	33.6
GMVD1 ->MultiViewX (70.3)							
UDA vanilla	69.1	80.7	**87.8**	86.4	81.5	67.4	50.7
UDA local-max	73.4	79.3	**87.8**	87.3	81.3	66.6	57.4
GMVD2 ->multiviewx (66.9)							
UDA vanilla	0.0	0.0	74.9	86.1	82.8	71.0	55.6
UDA local-max	79.9	84.1	**88.1**	86.1	80.1	70.5	51.6
multiviewx 1,2,3->4,5,6 (54.7)							
UDA vanilla	15.5	3.9	40.6	50.5	55.2	46.7	41.9
UDA local-max	58.1	**63.9**	63.1	61.3	56.3	48.3	39.6

Table 5 Performance of the baseline when evaluated using either vanilla or the proposed local-max pseudo-labeling

	MODA				MODP			
Method	$\tau = 0.2$	0.3	0.4	0.5	0.2	0.3	0.4	0.5
MultiviewX → Wildtrack								
Vanilla	0.0	42.9	70.0	**63.1**	71.9	72.7	73.6	75.0
Local-max	**42.0**	**59.2**	**70.1**	62.6	**72.4**	**73.2**	**73.9**	**75.1**
Wildtrack →MultiviewX								
Vanilla	25.0	35.9	32.5	24.7	64.2	66.4	67.1	68.6
Local-max	**48.5**	**41.5**	**33.2**	**24.8**	**66.0**	**67.5**	**68.2**	**69.3**
	Precision				Recall			
Method	0.2	0.3	0.4	0.5	0.2	0.3	0.4	0.5
MultiviewX → Wildtrack								
Vanilla	40.0	66.0	89.2	97.3	**95.5**	**88.2**	**79.6**	**64.9**
Local-max	**65.9**	**78.1**	**92.4**	**97.6**	87.0	82.4	76.4	64.2
Wildtrack →MultiviewX								
Vanilla	63.9	82.8	92.6	95.6	**57.2**	**45.2**	**35.3**	**25.9**
Local-max	**95.1**	**98.6**	**99.0**	**98.9**	51.1	42.1	33.5	25.1

Pseudo-labeling: Table 4 shows MODA of our UDA method using either vanilla pseudo-labeling or local-max pseudo-labeling. For convenience, we show the MODA of the baseline (from Tables 1 and 2) in parenthesis in the benchmark headings. It can be seen that UDA local-max yields the highest MODA in six out of seven benchmarks and offers improvements over the baseline across a wider range of τ values. Additionally, while self-training tend to diverge to increasing amounts of false positives when τ is set too low, often leading to zero MODA in the end of training, the rate of divergence is much faster for the vanilla method on most benchmarks. Meanwhile, local-max is more robust to low values of τ, mitigating the risk of degrading the performance of the baseline significantly. In practice, using the same τ on all benchmarks is feasible. Specifically, $\tau = 0.30$ is well-suited for local-max, while $\tau = 0.40$ works best for

Table 6 Performance comparison (MODA) of self-training using local-max pseudo-labeling with different values of k_d

k_d=	1	2	3	5	7
MultiviewX → Wildtrack (70.0)					
	81.2	80.9	79.9	78.3	–
GMVD1 →MultiviewX (70.3)					
	86.9	**88.1**	88.0	87.8	86.4

Table 8 Performance comparison (MODA) of self-training using different combinations of data augmentation

w/o	DV	MVA	3DR	All	DV+3DR
MultiviewX → Wildtrack (70.0)					
76.8	79.7	80.8	**85.0**	81.8	84.7
Wildtrack → MultiviewX (35.9)					
73.1	77.4	76.0	79.8	80.7	**82.4**
Wildtrack 2,4,5,6 → 1,3,5,7 (75.2)					
78.0	79.3	**79.4**	79.2	79.0	**79.4**
Wildtrack 1,3,5,7 → 2,4,5,6 (72.3)					
79.9	**81.9**	80.6	79.5	80.0	81.4
MultiviewX 1,2,6→3,4,5 (54.7)					
62.9	63.6	**65.1**	63.3	62.6	64.2
GMVD1 → MultiviewX (70.3)					
88.0	88.3	87.1	88.8	87.0	**89.0**
GMVD2 → MultiviewX (66.9)					
87.9	87.8	87.7	**89.1**	87.4	88.8

the vanilla method. Under these settings, local-max outperforms vanilla on six out of seven benchmarks.

It is also noteworthy that $\tau = 0.25$ yields the best result on two benchmarks, suggesting that our rough selection of $\tau \in \{0.2, 0.3, 0.4\}$ used throughout the paper leaves room for improvement.

To further analyze vanilla and local-max pseudo-labeling, we evaluate the baseline model using either of the two methods for post-processing. Table 5 shows the results on MultiviewX→Wildtrack and Wildtrack→MultiviewX for different thresholds τ. It can be seen that local-max attains higher precision and MODP in all cases, demonstrating that detections that are local maxima are typically more reliable. However, recall is higher for the vanilla method, owing to the fact that it usually produces a larger number of detections. It is noteworthy that the difference between the two methods is more pronounced for small values of τ. This is because vanilla post-processing produces many detections that are not local maxima in this case. Since these detections are less reliable, our method attains much higher MODA in this regime. Consequently, our method is able to harness reliable pseudo-labels at lower confidence levels, which evidently is particularly beneficial on the Wildtrack→MultiviewX benchmark.

To further analyze vanilla and local-max pseudo-labeling, we evaluate the baseline model using either of the two methods for post-processing. Table 5 shows the results on MultiviewX→Wildtrack and Wildtrack→ MultiviewX for different thresholds τ. It can be seen that local-max attains higher precision and MODP in all cases, demonstrating that detections that are local maxima are typically more reliable. However, recall is higher for the vanilla method, owing to the fact that it usually produces a larger number of detections. It is noteworthy that the difference between the two methods is more pronounced for small values of τ. This is because vanilla post-processing produces many detections that are not local maxima in this case. Since these detections are less reliable, our method attains much higher MODA in this regime. Consequently, our method is able to harness reliable pseudo-labels at lower confidence levels, which evidently is particularly beneficial on the Wildtrack→MultiviewX benchmark.

To further validate the robustness of our UDA method, we analyze its performance across different values of the parameter k_d on two benchmarks in Table 6. It is noteworthy that the considered neighborhood for local-max pseudo-labeling, defined in Eq. (5), is a square of size 70x70 cm when $k_d = 3$ since each cell in the predicted occupancy map corresponds to 10x10 cm. Hence, $k_d = 3$ is the largest value for which the entire square is within a radius of 0.5 ms, which is the distance used in NMS by conventional methods. In Table 6, it can be seen that our method works well as long as k_d is sufficiently small. Interestingly, the method works well even with the smallest possible value of $k_d = 1$. One could perhaps expect that the method would produce many false positives due to noise in the predictions with such a small kernel. Conversely, the predictions exhibit a reasonable smoothness that mitigates this problem, adding to the robustness of our method. Since k_d acts as a lower bound on the distance between any two pseudo-labels, a too large k_d risks degrading performance since it may introduce false negatives in crowded scenes, which happens around $k_d = 7$ on MultiviewX→Wildtrack.

Mean teacher α: Table 7 show the performance in MODA of our UDA method when trained for either 5 or 20

Table 7 Performance comparison (MODA) of self-training for 5 or 20 epochs using different values for α

Epochs	α=	0	0.9	0.99	0.999	1
MultiviewX → Wildtrack (70.0)						
5		–	–	79.7	76.3	77.2
20		–	–	79.1	**81.2**	77.3
GMVD1 →MultiviewX (70.3)						
5		85.3	88.0	**88.2**	83.5	79.0
20		86.8	87.9	87.8	85.3	79.2

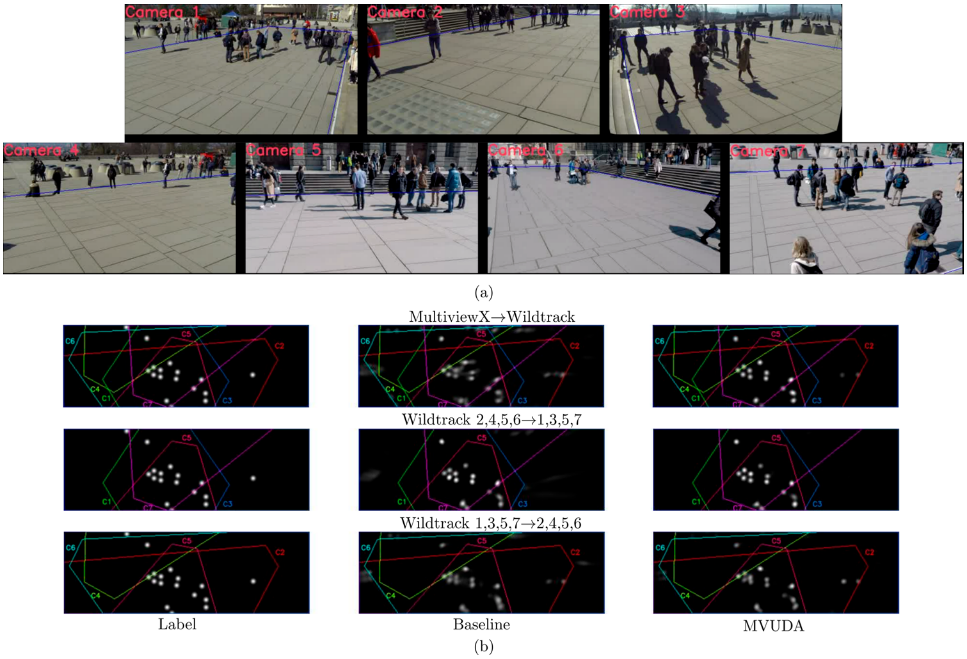
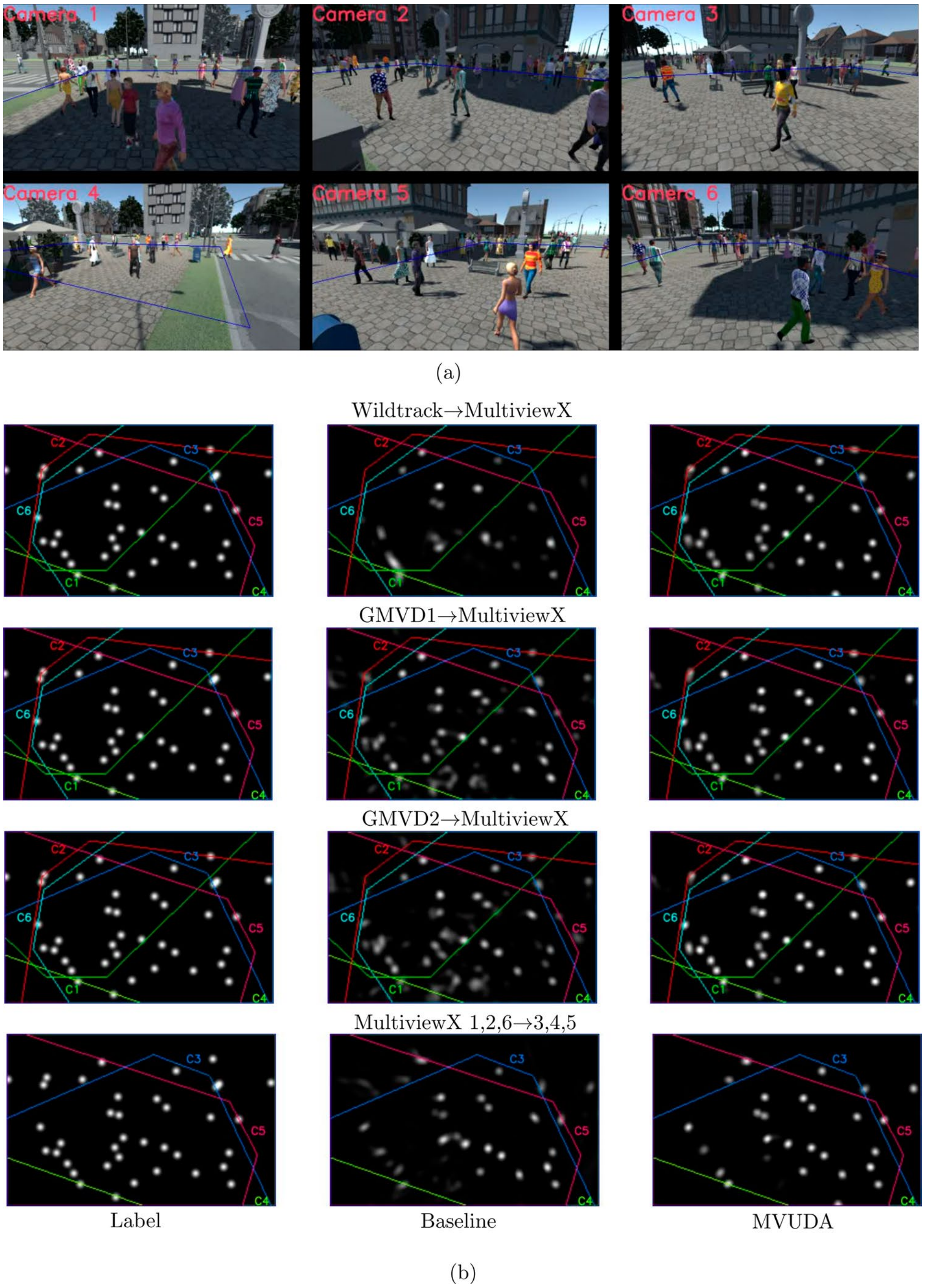


Fig. 3 A test sample from Wildtrack (**a**), as well as the associated label and predictions of the baseline and MVUDA (**b**). The predictions are produced by the methods trained on the specified benchmark, hence the results differ between the rows. Note that the label is identical across all rows since it is associated with the same test sample (**a**) in all benchmarks, although only a subset of the available cameras is used in the cases Wildtrack 2,4,5,6→1,3,5,7 and Wildtrack 1,3,5,7→2,4,5,6

epochs with different values of the parameter α. Note that $\alpha = 0$ implies that the teacher model equals the student (i.e., the student model is creating pseudo-labels), while $\alpha = 1$ implies that the frozen baseline model creates the pseudo-labels throughout training. It can be seen that both $\alpha = 0.99$ and $\alpha = 0.999$ yields decent performance on both benchmarks when training for 5 and 20 epochs, although a slowly evolving teacher ($\alpha = 0.999$) seems to benefit from longer trainings. We also note that a too low value of α leads to stability issues on one benchmark, owing to the rapid updates of the teacher model. Moreover, while freezing the teacher with $\alpha = 1$ works reasonable well for both benchmarks, it doesn't yield the best performance since it misses the opportunity to improve the quality of the pseudo-labels as training progresses. For additional experiments, please refer to the supplementary material.

Data augmentation Since data augmentation is an essential ingredient in self-training, we investigate three different methods that recently have been proposed for multi-view pedestrian detection. In Table 8, we present experiments with Dropview (DV) [6], 3D random occlusion (3DR) [8], and the two-level data augmentation developed in MVAug (MVA) [9]. It can be seen that each of these augmentation methods increases performance on most benchmarks. However, when combining the different methods, the best performance is achieved by DV and 3DR (excluding MVA). Similar results were obtained when we studied the generalization capability of the baseline, for which experiments are provided in the supplementary material. Given the good performance of MVAug presented by [9], these results are a bit surprising. However, it is also convenient since MVAug is substantially more complex than the other two methods. This is because MVAug, unlike Dropview and 3DR, not only augments the input image, but also augments the perspective transformation applied to the features.

Qualitative results We further compare the predictions of MVUDA and the baseline from Tables 1 and 2 qualitatively on a few examples. Fig. 3 shows a test sample from the Wildtrack dataset and the associated label and predictions produced by the baseline and MVUDA for different benchmarks. To ease comparison, we visualize the raw predictions (before any post-processing) and the label after *softened* with a Gaussian kernel. It can be seen that MVUDA improves on the baseline mainly in two aspects: first, by reducing the predicted scores in regions where there are no pedestrians, and second, by producing more distinct detections that are not smeared out spatially. Similarly, Fig. 4 shows a test sample from the MultiviewX dataset and the associated label and predictions. In this case, MVUDA not only exhibits the aforementioned improvements, but also detects some pedestrians that are missed completely by the baseline.



(a)

Wildtrack→MultiviewX

GMVD1→MultiviewX

GMVD2→MultiviewX

MultiviewX 1,2,6→3,4,5

Label

Baseline

MVUDA

(b)

Fig. 4 A test sample from MultiViewX (**a**), as well as the associated label and predictions of the baseline and MVUDA (**b**). The predictions are produced by the methods trained on the specified benchmark, hence the results differ between the rows. Note that the label is identical across all rows since it is associated with the same test sample (**a**) in all benchmarks, although only cameras 3,4,5 are used for testing in MultiviewX 1,2,6→3,4,5

4 Conclusions

In this paper, we presented MVUDA, the first unsupervised domain adaptive (UDA) method for multi-view pedestrian detection that eliminates the need for auxiliary labeled datasets. Our approach leverages mean teacher self-training with a novel pseudo-labeling method tailored for multi-view detection, significantly increasing pseudo-label reliability and the effectiveness of the overall framework. Extensive experiments demonstrate the efficacy of our method and motivate key design choices. By reducing the reliance on labeled data and achieving superior performance, we believe MVUDA sets a strong baseline for future research in unsupervised domain adaptation and holds significant potential for real-world applications.

Acknowledgements This work is supported by AB Volvo, the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation, and the Vinnova funded project SMILE IV (2023-00789). The experiments were enabled by resources provided by the National Academic Infrastructure for Supercomputing in Sweden (NAISS), partially funded by the Swedish Research Council through grant agreement no. 2022-06725.

Author Contributions E.B. Methodology development, Data curation, Software, Experiments, Writing manuscript, Preparation of figures L.S. Methodology development, Supervision, Writing - review and editing K.B. Methodology development, Supervision, Project administration, Writing - review and editing K.A. Methodology development, Supervision, Writing - review and editing

Funding Open access funding provided by Chalmers University of Technology.

Data Availability The code is available at https://github.com/ErikBrorsson/MVUDA. Data: Wildtrack https://www.epfl.ch/labs/cvlab/data/data-wildtrack MultiviewX https://github.com/hou-yz/MVDet GMVD https://github.com/jeetv/GMVD dataset

Declarations

Conflict of interest The authors have no Conflict of interest to declare that are relevant to the content of this article.

Conflict of interest The authors declare no Conflict of interest.

References

1. Ferryman, J., Shahrokni, A.: Pets2009: Dataset and challenge. In: 2009 Twelfth IEEE International Workshop on Performance Evaluation of Tracking and Surveillance, pp. 1–6 (2009). IEEE
2. Coates, A., Ng, A.Y.: Multi-camera object detection for robotics. In: 2010 IEEE International Conference on Robotics and Automation, pp. 412–419 (2010). IEEE
3. Ren, J., Xu, M., Orwell, J., Jones, G.A.: Multi-camera video surveillance for real-time analysis and reconstruction of soccer games. Mach. Vis. Appl. **21**, 855–863 (2010)
4. Zhang, Z., Hajieghrary, H., Dean, E., Åkesson, K.: Prescient collision-free navigation of mobile robots with iterative multimodal motion prediction of dynamic obstacles. IEEE Robotics Autom. Lett. **8**(9), 5488–5495 (2023). https://doi.org/10.1109/LRA.2023.3296333
5. Hou, Y., Zheng, L., Gould, S.: Multiview detection with feature perspective transformation. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) Computer Vision - ECCV 2020, pp. 1–18. Springer, Cham (2020)
6. Vora, J., Dutta, S., Jain, K., Karthik, S., Gandhi, V.: Bringing generalization to deep multi-view pedestrian detection. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 110–119 (2023)
7. Song, L., Wu, J., Yang, M., Zhang, Q., Li, Y., Yuan, J.: Stacked homography transformations for multi-view pedestrian detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 6049–6057 (2021)
8. Qiu, R., Xu, M., Yan, Y., Smith, J.S., Yang, X.: 3D random occlusion and multi-layer projection for deep multi-camera pedestrian localization. In: European Conference on Computer Vision, pp. 695–710 (2022). Springer
9. Engilberge, M., Shi, H., Wang, Z., Fua, P.: Two-level data augmentation for calibrated multi-view detection. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 128–136 (2023)
10. Aung, S., Park, H., Jung, H., Cho, J.: Enhancing multi-view pedestrian detection through generalized 3D feature pulling. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 1196–1205 (2024)
11. Deng, J., Li, W., Chen, Y., Duan, L.: Unbiased mean teacher for cross-domain object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4091–4101 (2021)
12. Hoyer, L., Dai, D., Van Gool, L.: Daformer: Improving network architectures and training strategies for domain-adaptive semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9924–9935 (2022)
13. Li, Y.-J., Dai, X., Ma, C.-Y., Liu, Y.-C., Chen, K., Wu, B., He, Z., Kitani, K., Vajda, P.: Cross-domain adaptive teacher for object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7581–7590 (2022)
14. Tarvainen, A., Valpola, H.: Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. Advances in neural information processing systems **30** (2017)
15. Lima, J.P., Thomas, D., Uchiyama, H., Teichrieb, V.: Toward unlabeled multi-view 3D pedestrian detection by generalizable AI: techniques and performance analysis. In: 2023 36th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI), pp. 1–6 (2023). IEEE
16. Lima, J.P., Thomas, D., Uchiyama, H., Teichrieb, V.: Mean teacher for unsupervised domain adaptation in multi-view 3D pedestrian detection. In: 2024 37th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI), pp. 1–6 (2024). IEEE
17. Fleuret, F., Berclaz, J., Lengagne, R., Fua, P.: Multicamera people tracking with a probabilistic occupancy map. IEEE Trans. Pattern Anal. Mach. Intell. **30**(2), 267–282 (2007)
18. Alahi, A., Jacques, L., Boursier, Y., Vandergheynst, P.: Sparsity driven people localization with a heterogeneous network of cameras. J. Mathem. Imaging Vision **41**, 39–58 (2011)
19. Peng, P., Tian, Y., Wang, Y., Li, J., Huang, T.: Robust multiple cameras pedestrian detection with multi-view bayesian network. Pattern Recogn. **48**(5), 1760–1772 (2015)
20. Lima, J.P., Roberto, R., Figueiredo, L., Simões, F., Thomas, D., Uchiyama, H., Teichrieb, V.: 3D pedestrian localization using multiple cameras: A generalizable approach. Mach. Vis. Appl. **33**(4), 61 (2022)
21. López-Cifuentes, A., Escudero-Viñolo, M., Bescós, J., Carballeira, P.: Semantic-driven multi-camera pedestrian detection. Knowl. Inf. Syst. **64**(5), 1211–1237 (2022)

22. Roig, G., Boix, X., Shitrit, H.B., Fua, P.: Conditional random fields for multi-camera object detection. In: 2011 International Conference on Computer Vision, pp. 563–570 (2011). IEEE
23. Qiu, R., Xu, M., Yan, Y., Smith, J.S., Ling, Y.: PPM: A boolean optimizer for data association in multi-view pedestrian detection. Pattern Recogn. **156**, 110807 (2024)
24. Chavdarova, T., Fleuret, F.: Deep multi-camera people detection. In: 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA), pp. 848–853 (2017). IEEE
25. Baqué, P., Fleuret, F., Fua, P.: Deep occlusion reasoning for multi-camera multi-target detection. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 271–279 (2017)
26. Lee, W.-Y., Jovanov, L., Philips, W.: Multi-view target transformation for pedestrian detection. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 90–99 (2023)
27. Teepe, T., Wolters, P., Gilg, J., Herzog, F., Rigoll, G.: EarlyBird: Early-fusion for multi-view tracking in the bird's eye view. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 102–111 (2024)
28. Hou, Y., Zheng, L.: Multiview detection with shadow transformer (and view-coherent data augmentation). In: Proceedings of the 29th ACM International Conference on Multimedia, pp. 1673–1682 (2021)
29. Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., March, M., Lempitsky, V.: Domain-adversarial training of neural networks. J. Mach. Learn. Res. **17**(59), 1–35 (2016)
30. Long, M., Cao, Y., Wang, J., Jordan, M.: Learning transferable features with deep adaptation networks. In: International Conference on Machine Learning, pp. 97–105 (2015). PMLR
31. Saito, K., Watanabe, K., Ushiku, Y., Harada, T.: Maximum classifier discrepancy for unsupervised domain adaptation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3723–3732 (2018)
32. Hoffman, J., Wang, D., Yu, F., Darrell, T.: Fcns in the wild: Pixel-level adversarial and constraint-based adaptation. arXiv preprint arXiv:1612.02649 (2016)
33. Hoffman, J., Tzeng, E., Park, T., Zhu, J.-Y., Isola, P., Saenko, K., Efros, A., Darrell, T.: Cycada: Cycle-consistent adversarial domain adaptation. In: International Conference on Machine Learning, pp. 1989–1998 (2018). Pmlr
34. Gong, R., Li, W., Chen, Y., Gool, L.V.: Dlow: Domain flow for adaptation and generalization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2477–2486 (2019)
35. Tsai, Y.-H., Hung, W.-C., Schulter, S., Sohn, K., Yang, M.-H., Chandraker, M.: Learning to adapt structured output space for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7472–7481 (2018)
36. Cai, Q., Pan, Y., Ngo, C.-W., Tian, X., Duan, L., Yao, T.: Exploring object relation in mean teacher for cross-domain detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 11457–11466 (2019)
37. Cao, S., Joshi, D., Gui, L.-Y., Wang, Y.-X.: Contrastive mean teacher for domain adaptive object detectors. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 23839–23848 (2023)
38. Lee, D.-H., : Pseudo-Label: The simple and efficient semi-supervised learning method for deep neural networks. In: Workshop on Challenges in Representation Learning, ICML, vol. 3, p. 896 (2013). Atlanta
39. Zheng, Z., Yang, Y.: Rectifying pseudo label learning via uncertainty estimation for domain adaptive semantic segmentation. Int. J. Comput. Vision **129**(4), 1106–1120 (2021)
40. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
41. Chavdarova, T., Baqué, P., Bouquet, S., Maksai, A., Jose, C., Bagautdinov, T., Lettry, L., Fua, P., Van Gool, L., Fleuret, F.: Wildtrack: A multi-camera HD dataset for dense unscripted pedestrian detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5030–5039 (2018)
42. Kasturi, R., Goldgof, D., Soundararajan, P., Manohar, V., Garofolo, J., Bowers, R., Boonstra, M., Korzhova, V., Zhang, J.: Framework for performance evaluation of face, text, and vehicle detection and tracking in video: Data, metrics, and protocol. IEEE Trans. Pattern Anal. Mach. Intell. **31**(2), 319–336 (2008)
43. Smith, L.N., Topin, N.: Super-convergence: Very fast training of neural networks using large learning rates. In: Artificial Intelligence and Machine Learning for Multi-domain Operations Applications, vol. 11006, pp. 369–386 (2019). SPIE
44. Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255 (2009). IEEE
45. Leal-Taixé, L., Milan, A., Reid, I., Roth, S., Schindler, K.: MOTChallenge 2015: Towards a Benchmark for Multi-Target Tracking (2015). https://arxiv.org/abs/1504.01942
46. Qiu, R., Xu, M., Ling, Y., Smith, J.S., Yan, Y., Wang, X.: A deep top-down framework towards generalisable multi-view pedestrian detection. Neurocomputing **607**, 128458 (2024)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Erik Brorsson received his M.Sc. degree in Complex Adaptive Systems from Chalmers University of Technology, Sweden, in 2021. He is currently an industrial Ph.D. student with AB Volvo and the Department of Electrical Engineering at Chalmers University of Technology. His research interests include camera-based perception, such as semantic segmentation and object detection, learning with limited labeled data, and multi-view perception.

Lennart Svensson is a Professor of signal processing with Chalmers University of Technology. His research interests include machine learning and Bayesian inference in general, and nonlinear filtering, deep learning, and tracking in particular. He has organized a massive open online course on multiple object tracking. He was the recipient of best paper awards at the International Conference on Information Fusion in 2009, 2010, 2017, and 2019.

Kristofer Bengtsson received a Ph.D. degree in signals and systems from the Chalmers University of Technology, Gothenburg, Sweden, in 2012. From 2001 to 2005, he was with Advanced Flow Control AB, developing control systems and user interfaces, and from 2005 to 2011, he was with Teamster AB, Gothenburg, an automation firm. He was an Associate Professor in the automation research group at Chalmers until 2022 and is now a senior researcher in Smart and Connected operations at Volvo Group. His current research interest includes using AI in real applications, OT/IT convergence, computer vision, robotics, intelligent automation, and how to implement and run algorithms for prediction, planning, and optimization in practice.

Knut Åkesson is a Professor of Automation at Chalmers University of Technology. His research centers on rigorous methods for the analysis and verification of cyber-physical systems, as well as planning and control of large-scale mobile robot fleets. His recent work increasingly incorporates deep machine-learning–based perception and motion forecasting. He is particularly interested in integrating modern perception and learning methods with classical optimization-based approaches to enable large-scale, fully autonomous systems that are both safe and efficient.