Article

Co-Training for Deep Object Detection: Comparing Single-Modal and Multi-Modal Approaches

Jose L. Gómez [1,2,*], Gabriel Villalonga [1] and Antonio M. López [1,2]

[1] Computer Vision Center (CVC), Universitat Autònoma de Barcelona (UAB), 08193 Bellaterra, Spain; gvillalonga@cvc.uab.es (G.V.); antonio@cvc.uab.es (A.M.L.)
[2] Computer Science Department, Universitat Autònoma de Barcelona (UAB), 08193 Bellaterra, Spain
[*] Correspondence: jlgomez@cvc.uab.es

Abstract: Top-performing computer vision models are powered by convolutional neural networks (CNNs). Training an accurate CNN highly depends on both the raw sensor data and their associated ground truth (GT). Collecting such GT is usually done through human labeling, which is time-consuming and does not scale as we wish. This data-labeling bottleneck may be intensified due to domain shifts among image sensors, which could force per-sensor data labeling. In this paper, we focus on the use of co-training, a semi-supervised learning (SSL) method, for obtaining self-labeled object bounding boxes (BBs), i.e., the GT to train deep object detectors. In particular, we assess the goodness of multi-modal co-training by relying on two different views of an image, namely, appearance (RGB) and estimated depth (D). Moreover, we compare appearance-based single-modal co-training with multi-modal. Our results suggest that in a standard SSL setting (no domain shift, a few human-labeled data) and under virtual-to-real domain shift (many virtual-world labeled data, no human-labeled data) multi-modal co-training outperforms single-modal. In the latter case, by performing GAN-based domain translation both co-training modalities are on par, at least when using an off-the-shelf depth estimation model not specifically trained on the translated images.

Keywords: co-training; multi-modality; vision-based object detection; ADAS; self-driving

1. Introduction

Supervised deep learning enables accurate computer vision models. Key for this success is the access to raw sensor data (i.e., images) with ground truth (GT) for the visual task at hand (e.g., image classification [1], object detection [2] and recognition [3], pixel-wise instance/semantic segmentation [4,5], monocular depth estimation [6], 3D reconstruction [7], etc.). The supervised training of such computer vision models, which are based on convolutional neural networks (CNNs), is known to required very large amounts of images with GT [8]. While, until one decade ago, acquiring representative images was not easy for many computer vision applications (e.g., for onboard perception), nowadays, the bottleneck has shifted to the acquisition of the GT. The reason is that this GT is mainly obtained through human labeling, whose difficulty depends on the visual task. In increasing order of labeling time, we see that image classification requires image-level tags, object detection requires object bounding boxes (BBs), instance/semantic segmentation requires pixel-level instance/class silhouettes, and depth GT cannot be manually provided. Therefore, manually collecting such GT is time-consuming and does not scale as we wish. Moreover, this data labeling bottleneck may be intensified due to domain shifts among different image sensors, which could drive to per-sensor data labeling.

To address the curse of labeling, different meta-learning paradigms are being explored. In self-supervised learning (SfSL) the idea is to train the desired models with the help of auxiliary tasks related to the main task. For instance, solving automatically generated jigsaw puzzles helps to obtain more accurate image recognition models [9], while stereo and structure-from-motion (SfM) principles can provide self-supervision to train monocular

depth estimation models [10]. In active learning (AL) [11,12], there is a human—model collaborative loop, where the model proposes data labels, known as pseudo-labels, and the human corrects them so that the model learns from the corrected labels too; thus, aiming at a progressive improvement of the model accuracy. In contrast to AL, semi-supervised learning (SSL) [13,14] does not require human intervention. Instead, it is assumed the availability of a small set of off-the-shelf labeled data and a large set of unlabeled data, and both datasets must be used to obtain a more accurate model than if only the labeled data were used. In SfSL, the model trained with the help of the auxiliary tasks is intended to be the final model of interest. In AL and SSL, it is possible to use any model with the only purpose of self-labeling the data, i.e., producing the pseudo-labels, and then use labels and pseudo-labels for training the final model of interest.

In this paper we focus on co-training [15,16], a type of SSL algorithm. Co-training self-labels data through the mutual improvement of two models. These models analyze the unlabeled data according to their different views of these data. Our work focuses on onboard vision-based perception for driver assistance and autonomous driving. In this context, vehicle and pedestrian detection are key functionalities. Accordingly, we apply co-training to significantly reduce human intervention when labeling these objects (in computer vision terminology) for training the corresponding deep object detector. Therefore, the labels are BBs locating the mentioned traffic participants in the onboard images. More specifically, we consider two settings. On the one hand, as is usual in SSL, we assume the availability of a small set of human-labeled images (i.e., with BBs for the objects of interests), and a significantly larger set of unlabeled images. On the other hand, we do not assume human labeling at all, but we have a set of virtual-world images with automatically generated BBs.

This paper is the natural continuation of the work presented by Villalonga & López [17]. In this previous work, a co-training algorithm for deep object detection is presented, addressing the two above-mentioned settings too. In [17], the two views of an image consist of the original RGB representation and its horizontal mirror; thus, it is a single-modal co-training based on appearance. However, a priori, the higher difference among data views the more accurate pseudo-labels can be expected from co-training. Therefore, as a major novelty of this paper, we explore the use of two image modalities in the role of co-training views. In particular, one view is the appearance (i.e., the original RGB), while the other view is the corresponding depth (D) as estimated by a state-of-the-art monocular depth estimation model [18]. Thus, we term this approach as multi-modal co-training; however, it can still be considered a single-sensor because still relies only on RGB images. Figure 1 illustrates these different views for images that we use in our experiments.

In this setting, the research questions that we address are two: (Q1) Is multi-modal (RGB/D) co-training effective on the task of providing pseudo-labeled object BBs?; (Q2) How does perform multi-modal (RGB/D) co-training compared to single-modal (RGB)?. After adapting the method presented in [17] to work with both, the single and the multi-modal data views, we ran a comprehensive set of experiments for answering these two questions. Regarding (Q1), we conclude that, indeed, multi-modal co-training is rather effective. Regarding (Q2), we conclude that in a standard SSL setting (no domain shift, a few human-labeled data) and under virtual-to-real domain shift (many virtual-world labeled data, no human-labeled data) multi-modal co-training outperforms single-modal. In the latter case, when GAN-based virtual-to-real image translation is performed [19] (i.e., as image-level domain adaptation) both co-training modalities are on par; at least, by using an off-the-shelf monocular depth estimation model not specifically trained on the translated images.

We organize the rest of the paper as follows. Section 2 reviews related works. Section 3 draws the co-training algorithm. Section 4 details our experimental setting, discussing the obtained results in terms of (Q1) and (Q2). Section 5 summarizes the presented work, suggesting lines of continuation.

Figure 1. From top to bottom: samples from KITTI ($\mathcal{K}$), Waymo ($\mathcal{W}$), and Virtual-world ($\mathcal{V}$) datasets. Middle column: cropped patch from an original image. Left column: horizontal mirror of the original patch. Right column: monocular depth estimation [18] from the original patch. Left-middle columns are the views used for co-training in [17]. Right-middle columns are the views also used in this paper.

2. Related Work

As we have mentioned before, co-training falls in the realm of SSL. Thus, here we summarize previous related works applying SSL methods. The input to these methods consists of a labeled dataset, $\mathcal{X}^l$, and an unlabeled one, $\mathcal{X}^u$, with $\#\mathcal{X}^u \gg \#\mathcal{X}^l$ and $\mathcal{D}_{\mathcal{X}^u} = \mathcal{D}_{\mathcal{X}^l}$, where $\#\mathcal{X}$ is the cardinality of the set $\mathcal{X}$ and $\mathcal{D}_{\mathcal{X}}$ refers to the domain from which $\mathcal{X}$ has been drawn. Note that, when the latter requirement does not hold, we are under a domain shift setting. The goal of a SSL method is to use both $\mathcal{X}^l$ and $\mathcal{X}^u$ to allow the training of a predictive model, ϕ, so that its accuracy is higher than if only $\mathcal{X}^l$ is used for its training. In other words, the goal is to leverage unlabeled data.

A classical SSL approach is the so-called self-training, introduced by Yarowsky [20] in the context of language processing. Self-training is an incremental process that starts by training ϕ on $\mathcal{X}^l$; then, ϕ runs on $\mathcal{X}^u$, and its predictions are used to form a pseudo-labeled set $\mathcal{X}^{\hat{l}}$, further used together with $\mathcal{X}^l$ to retrain ϕ. This is repeated until convergence, and the accuracy of ϕ, as well as the quality of $\mathcal{X}^{\hat{l}}$, are supposed to become higher as the cycles progress. Jeong et al. [21] used self-training for deep object detection (on PASCAL VOC and MS-COCO datasets). To collect $\mathcal{X}^{\hat{l}}$, a consistency loss is added while training ϕ, which is a CNN for object detection in this case, together with a mechanism for removing predominant backgrounds. The consistency loss is based on the idea that $\phi(\mathrm{I}^u) \sim \phi(\mathrm{I}^{u\,\neg})^{\,\neg}$, where I^u is an unlabeled image, and "$\neg$" refers to performing horizontal mirroring. Lokhande et al. [22] used self-training for deep image classification. In this case, the original activation functions of ϕ, a CNN for image classification, must be changed to Hermite polynomials. Note that these two examples of self-training involve modifications either in the architecture of ϕ [22] or in its training framework [21]. However, we aim at using a given ϕ together with its training framework as a black box, so performing SSL only at the data level. In this way, we can always benefit from state-of-the-art models and training frameworks, i.e., avoiding changing the SSL approach if those change. In this way, we can also decouple the model used to produce pseudo-labels from the model that would be trained with them for deploying the application of interest.

A major challenge when using self-training is to avoid drifting to erroneous pseudo-labels. Note that, if $\mathcal{X}^{\hat{l}}$ is biased to some erroneous pseudo-labels, when using this set to

retrain ϕ incrementally, a point can be reached where $\mathcal{X}^l$ cannot compensate the errors in $\mathcal{X}^{\hat{l}}$, and ϕ may end learning wrong data patterns and so producing more erroneous pseudo-labels. Thus, as alternative to the self-training of Yarowsky [20], Blum and Mitchell proposed co-training [15]. Briefly, co-training is based on two models, ϕ_{v_1} and ϕ_{v_2}, each one incrementally trained on different data features, termed as views. In each training cycle, ϕ_{v_1} and ϕ_{v_2} collaborate to form $\mathcal{X}^{\hat{l}} = \mathcal{X}^{\hat{l}}_{v_1} \cup \mathcal{X}^{\hat{l}}_{v_2}$. Where, $\mathcal{X}^{\hat{l}}_{v_i}$ and $\mathcal{X}^l$ are used to retrain $\phi_{v_i}, i \in \{1, 2\}$. This is repeated until convergence. It is assumed that each view, v_i, is discriminant enough as to train an accurate ϕ_{v_i}. Different implementations of co-training, may differ in the collaboration policy. Our approach follows the disagreement idea introduced by Guz et al. [16] in the context of sentence segmentation, later refined by Tur [23] to address domain shifts in the context of natural language processing. In short, only pseudo-labels of high confidence for ϕ_{v_i} but of low confidence for ϕ_{v_j}, $i, j \in \{1, 2\}, i \neq j$, are considered as part of $\mathcal{X}^{\hat{l}}_{v_j}$ in each training cycle. Soon, disagreement-based SSL attracted much interest [24].

In general, ϕ_{v_1} and ϕ_{v_2} can be based on different data views by either training on different data samples ($\mathcal{X}^{\hat{l}}_{v_1} \neq \mathcal{X}^{\hat{l}}_{v_2}$) or being different models (e.g., ϕ_{v_1} and ϕ_{v_2} can be based on two different CNN architectures). The disagreement-based co-training falls in the former case. In this line, Qiao et al. [25] used co-training for deep image classification, where the two different views are achieved by training on mutually adversarial samples. However, this implies linking the training of the ϕ_{v_i}'s at the level of the loss function, while, as we have mentioned before, we want to use these models as black boxes.

The most similar work to this paper is the co-training framework that we introduced in [17] since we work on top of it. In [17], two single-modal views are considered. These consist of using ϕ_{v_1} to process the original images from $\mathcal{X}^u$ while using ϕ_{v_2} to process their horizontally mirrored counterparts, and analogously for $\mathcal{X}^l$. A disagreement-based collaboration is applied to form $\mathcal{X}^{\hat{l}}_{v_1}$ and $\mathcal{X}^{\hat{l}}_{v_2}$. Moreover, not only the setting where $\mathcal{X}^l$ is based on human labels is considered, but also when it is based on virtual-world data. In the latter case, a GAN-based virtual-to-real image translation [19] is used as pre-processing for the virtual-world images, i.e., before taking them for running the co-training procedure. Very recently, Díaz et al. [26] presented co-training for visual object recognition. In other words, the paper addresses a classification problem, while we address both localization and classification to perform object detection. While the different views proposed in [26] rely on self-supervision (e.g., forcing image rotations), here, these rely on data multi-modality. In fact, in our previous work [17], we used mirroring to force different data views, which can be considered as a kind of self-supervision too. Here, after adapting and improving the framework used in [17], we confront this previous setting to a new multi-modal single-sensor version (Algorithms 1 and Figure 2). We focus on the case where ϕ_{v_1} works with the original images while ϕ_{v_2} works with their estimated depth. Analyzing this setting is quite interesting because appearance and depth are different views of the same data.

To estimate depth, we need an out-of-the-shelf monocular depth estimation (MDE) model, so that we can keep the co-training as a single-sensor even being multi-modal. MDE can be based on either LiDAR supervision, or stereo/SfM self-supervision, or combinations; where, both LiDAR and stereo data, and SfM computations, are only required at training time, but not at testing time. We refer to [6] for a review on MDE state-of-the-art. In this paper, to isolate the multi-modal co-training performance assessment as much as possible from the MDE performance, we have chosen the top-performing supervised method proposed by Yin et al. [18].

Finally, we would like to mention that there are methods in the literature that may be confused with co-training, so it is worth introducing a clarification note. This is the case of the co-teaching proposed by Han et al. [27] and the co-teaching+ of Yu et al. [28]. These methods have been applied to deep image classification to handle noisy labels on $\mathcal{X}^l$. However, citing Han et al. [27], *co-training is designed for SSL, and co-teaching is for learning*

with noisy (ground truth) labels (LNL); as LNL is not a special case of SSL, we cannot simply translate co-training from one problem setting to another problem setting.
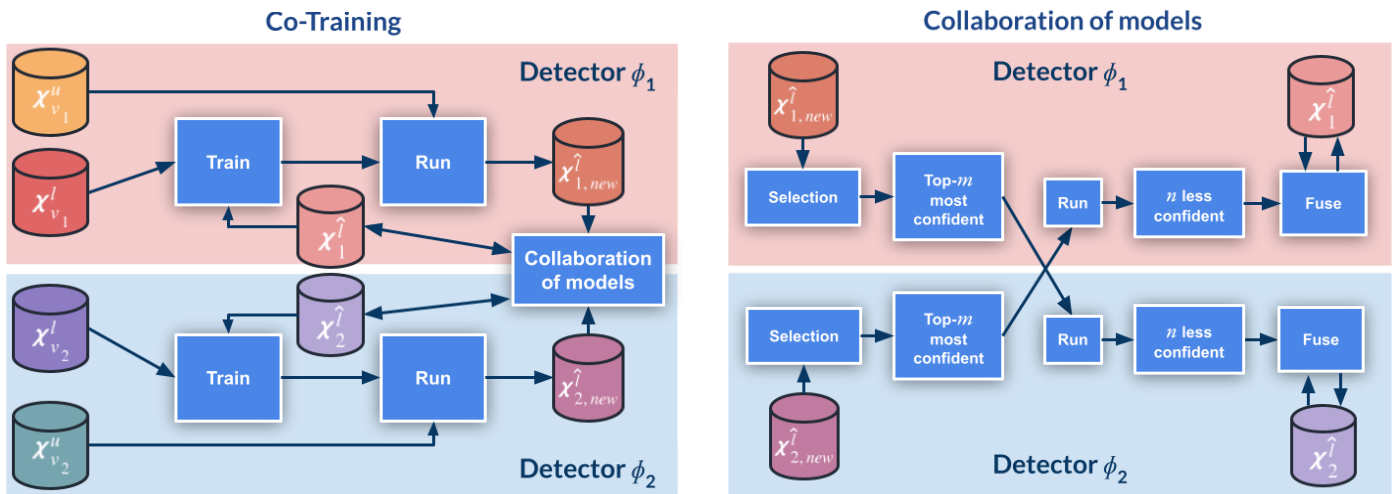

Figure 2. Co-training pipeline: the **left** diagram shows the global block structure, while the **right** diagram details the collaboration of models block. Symbols and procedures are based on Algorithms 1. We refer to this algorithm and the main text for a detailed explanation.

Algorithm 1: Self-labeling of object BBs by co-training.

Input : View-paired sets of labeled images: $\mathcal{X}_{v_1}^l, \mathcal{X}_{v_2}^l$
　　　　View-paired sets of unlabeled images: $\mathcal{X}_{v_1}^u, \mathcal{X}_{v_2}^u$
　　　　Object detection architecture, and its training hyper-parameters: $\Phi, \mathcal{H}_\Phi$
　　　　Co-training hyper-parameters: $\mathcal{H}_{ct} = \{T, N, n, m, \mathcal{H}_{stp}[, \mathcal{H}_{seq}]\}$

Output: New labeled images: $\mathcal{X}^{\hat{l}} \subseteq \mathcal{X}_{v_1}^u$

// **Initialize** models and working datasets.

$< \mathcal{X}_1^{\hat{l}}, \mathcal{X}_2^{\hat{l}}, k > \quad \leftarrow \quad < \varnothing, \varnothing, 0 >$

$\phi_1, \phi_2 \quad \leftarrow \quad \text{Train}(\Phi, \mathcal{H}_\Phi, \mathcal{X}_{v_1}^l, \mathcal{X}_1^{\hat{l}}), \text{Train}(\Phi, \mathcal{H}_\Phi, \mathcal{X}_{v_2}^l, \mathcal{X}_2^{\hat{l}})$

$\mathcal{X}_{1,new}^{\hat{l}}, \mathcal{X}_{2,new}^{\hat{l}} \quad \leftarrow \quad \text{Run}(\phi_1, \mathcal{X}_{v_1}^u, T), \text{Run}(\phi_2, \mathcal{X}_{v_2}^u, T)$

repeat

$\quad\quad \mathcal{X}_{old}^{\hat{l}} \quad \leftarrow \quad \mathcal{X}_{1,new}^{\hat{l}}$

$\quad\quad$ // **Collaboration** of models.

$\quad \mathcal{X}_{1,\uparrow}^{\hat{l}}, \mathcal{X}_{2,\uparrow}^{\hat{l}} \quad \leftarrow \quad \text{Slct}(\uparrow, m, \text{Rnd}(\mathcal{X}_{1,new}^{\hat{l}}, N[, \mathcal{H}_{seq}, k])), \text{Slct}(\uparrow, m, \text{Rnd}(\mathcal{X}_{2,new}^{\hat{l}}, N[, \mathcal{H}_{seq}, k]))$

$\quad \mathcal{X}_{1,\downarrow}^{\hat{l}}, \mathcal{X}_{2,\downarrow}^{\hat{l}} \quad \leftarrow \quad \text{Slct}(\downarrow, n, \text{Run}(\phi_1, \mathcal{X}_{2,\uparrow}^{\hat{l}}, T)), \text{Slct}(\downarrow, n, \text{Run}(\phi_2, \mathcal{X}_{1,\uparrow}^{\hat{l}}, T))$

$\quad \mathcal{X}_1^{\hat{l}}, \mathcal{X}_2^{\hat{l}} \quad \leftarrow \quad \text{Fuse}(\mathcal{X}_1^{\hat{l}}, \mathcal{X}_{1,\downarrow}^{\hat{l}}), \text{Fuse}(\mathcal{X}_2^{\hat{l}}, \mathcal{X}_{2,\downarrow}^{\hat{l}})$

$\quad\quad$ // **Retrain** models and **Update** working datasets.

$\quad\quad \phi_1, \phi_2 \quad \leftarrow \quad \text{Train}(\Phi, \mathcal{H}_\Phi, \mathcal{X}_{v_1}^l, \mathcal{X}_1^{\hat{l}}), \text{Train}(\Phi, \mathcal{H}_\Phi, \mathcal{X}_{v_2}^l, \mathcal{X}_2^{\hat{l}})$

$\quad \mathcal{X}_{1,new}^{\hat{l}}, \mathcal{X}_{2,new}^{\hat{l}} \quad \leftarrow \quad \text{Run}(\phi_1, \mathcal{X}_{v_1}^u, T), \text{Run}(\phi_2, \mathcal{X}_{v_2}^u, T)$

until $Stop?(\mathcal{H}_{stp}, \mathcal{X}_{old}^{\hat{l}}, \mathcal{X}_{1,new}^{\hat{l}}, k{+}{+})$

$\mathcal{X}^{\hat{l}} \quad \leftarrow \quad \mathcal{X}_{1,new}^{\hat{l}}$

return $\mathcal{X}^{\hat{l}}$

---

3. Method

In this section, we explain our co-training procedure with the support of Figure 2 and Algorithms 1. Up to a large extent, we follow the same terminology as in [17].

Input: The specific sets of labeled ($\mathcal{X}_{v_1}^l, \mathcal{X}_{v_2}^l$) and unlabeled ($\mathcal{X}_{v_1}^u, \mathcal{X}_{v_2}^u$) input data in Algorithms 1 determine if we are running on either a single or multi-modal setting. Also, if we are supported or not by virtual-world images or their virtual-to-real translated counterparts. Table 1, clarifies the different co-training settings depending on these datasets.

In Algorithms 1, view-paired sets means that each image of one set has a counterpart in the other, i.e., following Table 1, its horizontal mirror or its estimated depth. Since the co-training is agnostic to the specific object detector in use, we explicitly consider its corresponding CNN architecture, Φ, and training hyper-parameter, $\mathcal{H}_\Phi$, as inputs. Finally, $\mathcal{H}_{ct}$ consists of the co-training hyper-parameters, which we will introduce while explaining the part of the algorithm in which each of them is required.

Output: It consists in a set of images ($\mathcal{X}^{\hat{l}}$) from $\mathcal{X}^u_{v_1}$, for which co-training is providing pseudo-labels, i.e., object BBs in this paper. In our experiments, according to Table 1, $\mathcal{X}^u_{v_1}$ always corresponds to the unlabeled set of original real-world images. Since we consider as output a set of self-labeled images, which complement the input set of labeled images, they can be later used to train a model based on Φ or any other CNN architecture performing the same task (i.e., requiring the same type of BBs).

Initialize: First, the initial object detection models (ϕ_1, ϕ_2) are trained using the respective views of the labeled data ($\mathcal{X}^l_{v_1}, \mathcal{X}^l_{v_2}$). After their training, these models are applied to the respective views of the unlabeled data ($\mathcal{X}^u_{v_1}, \mathcal{X}^u_{v_2}$). Detections (i.e., object BBs) with a confidence over a threshold are considered pseudo-labels. Since we address a multi-class problem, per-class thresholds are contained in the set T, a hyper-parameter in $\mathcal{H}_{ct}$. The temporary self-labeled sets generated by ϕ_1 and ϕ_2 are $\mathcal{X}^{\hat{l}}_{1,new}$ and $\mathcal{X}^{\hat{l}}_{2,new}$, respectively. At this point no collaboration is produced between ϕ_1 and ϕ_2. In fact, while co-training loops (repeat body), the self-labeled sets resulting from the collaboration are $\mathcal{X}^{\hat{l}}_1$ and $\mathcal{X}^{\hat{l}}_2$, which are initialized as empty. In the training function, $\text{Train}(\Phi, \mathcal{H}_\Phi, \mathcal{S}^l, \mathcal{S}^{\hat{l}}) : \phi$, we use BB labels (in $\mathcal{S}^l$) and BB pseudo-labels (in $\mathcal{S}^{\hat{l}}$) indistinctly. However, we only consider background samples from $\mathcal{S}^l$, since, as co-training progresses, $\mathcal{S}^{\hat{l}}$ may be instantiated with a set of self-labeled images containing false negatives (i.e., undetected objects) which could be erroneously taken as hard negatives (i.e., background quite similar to objects) when training ϕ.

Collaboration: The two object detection models collaborate by exchanging pseudo-labeled images (Figure 2-right). This exchange is inspired in disagreement-based SSL [24]. Our specific approach is controlled by the co-training hyper-parameters N, n, m, and, in case of working with image sequences instead of with sets of isolated images, also by $\mathcal{H}_{seq} = \{\Delta t_1, \Delta t_2\}, \Delta t_1, \Delta t_2$. This approach consists of the following three steps.

(First step) Each model selects the set of its top-m most confident self-labeled images ($\mathcal{X}^{\hat{l}}_{1,\uparrow}, \mathcal{X}^{\hat{l}}_{2,\uparrow}$); where, the confidence of an image is defined as the average over the confidences of the pseudo-labels of the image, i.e., in our case, over the object detections. Thus, $\mathcal{X}^{\hat{l}}_{i,\uparrow} \subseteq \mathcal{X}^{\hat{l}}_{i,new}, i \in \{1, 2\}$. However, for creating $\mathcal{X}^{\hat{l}}_{i,\uparrow}$, we do not consider all the self-labeled images in $\mathcal{X}^{\hat{l}}_{i,new}$. Instead, to minimize bias and favor speed, we only consider N randomly selected images from $\mathcal{X}^{\hat{l}}_{i,new}$. In the case of working with image sequences, to favor variability in the pseudo-labels, the random choice is constrained to avoid using consecutive frames. This is controlled by thresholds Δt_1 and Δt_2; where Δt_1 controls the minimum frame distance between frames selected at the current co-training cycle (k), and Δt_2 among frames at current cycle with respect to frames selected in previous cycles ($< k$). We apply Δt_1 first, then Δt_2, and then the random selection among the frames passing these constraints.

(Second step) Model ϕ_i processes $\mathcal{X}^{\hat{l}}_{j,\uparrow}, i, j \in \{1, 2\}, i \neq j$, keeping the set of the n less confident self-labeled images for it. Thus, we obtain the new sets $\mathcal{X}^{\hat{l}}_{1,\downarrow}$ and $\mathcal{X}^{\hat{l}}_{2,\downarrow}$. Therefore, considering the first and second steps, we see that one model shares with the other those images that it has self-labeled with more confidence, and, of these, each model retains for retraining those that it self-labels with less confidence. Therefore, this step implements the actual collaboration between models ϕ_1 and ϕ_2.

(Third step) The self-labeled sets obtained in previous step ($\mathcal{X}^{\hat{l}}_{1,\downarrow}, \mathcal{X}^{\hat{l}}_{2,\downarrow}$) are fused with those accumulated from previous co-training cycles ($\mathcal{X}^{\hat{l}}_1, \mathcal{X}^{\hat{l}}_2$). This is done by properly calling the function $\text{Fuse}(\mathcal{S}^{\hat{l}}_{old}, \mathcal{S}^{\hat{l}}_{new}) : \mathcal{S}^{\hat{l}}$ for each view. The returned set of self-labeled

images, $\mathcal{S}^{\hat{l}}$, contains $\mathcal{S}^{\hat{l}}_{old} \cup \mathcal{S}^{\hat{l}}_{new} - \mathcal{S}^{\hat{l}}_{old} \cap \mathcal{S}^{\hat{l}}_{new}$, and, from $\mathcal{S}^{\hat{l}}_{old} \cap \mathcal{S}^{\hat{l}}_{new}$, only those self-labeled images in $\mathcal{S}^{\hat{l}}_{new}$ are added to $\mathcal{S}^{\hat{l}}$.

Retrain and update: At this point we have new sets of self-labeled images $(\mathcal{X}^{\hat{l}}_1, \mathcal{X}^{\hat{l}}_2)$, which, together with the corresponding input labeled sets $(\mathcal{X}^l_{v_1}, \mathcal{X}^l_{v_2})$, are used to retrain the models ϕ_1 and ϕ_2. Afterwards, these new models are used to obtain new temporary self-labeled set $(\mathcal{X}^{\hat{l}}_{1,new}, \mathcal{X}^{\hat{l}}_{2,new})$ through their application to the corresponding unlabeled sets $(\mathcal{X}^u_{v_1}, \mathcal{X}^u_{v_2})$. Then, co-training can start a new cycle.

Stop: The function $\text{Stop?}(\mathcal{H}_{stp}, \mathcal{S}^{\hat{l}}_{old}, \mathcal{S}^{\hat{l}}_{new}, k)$: Boolean determines if a new co-training cycle is executed. This is controlled by the co-training hyper-parameters $\mathcal{H}_{stp} = \{K_{min}, K_{max}, T_{\Delta_{mAP}}, \Delta K\}$. Co-training will execute a minimum of K_{min} cycles and a maximum of K_{max}, being k the current number. The parameters $\mathcal{S}^{\hat{l}}_{old}$ and $\mathcal{S}^{\hat{l}}_{new}$ are supposed to be instantiated with the sets of self-labeled images in previous and current co-training cycles, respectively. The similarity of these sets is monitored in each cycle, so that if its stable for more than ΔK consecutive cycles, convergence is assumed and co-trained stopped. This constrain could already be satisfied at $k = K_{min}$ provided $K_{min} \geqslant \Delta K$. The metric used to compute the similarity between these self-labeled sets is mAP (mean average precision) [29], where $\mathcal{S}^{\hat{l}}_{old}$ plays the role of GT and $\mathcal{S}^{\hat{l}}_{new}$ the role of results under evaluation. Then, mAP is considered stable between two consecutive cycles if its magnitude variation is below the threshold $T_{\Delta_{mAP}}$.

Table 1. The different configurations that we consider for Algorithms 1 in this paper, according to the input datasets. In the single-modal cases, we work only with RGB images (appearance), either from a real-world dataset ($\mathcal{R}_{RGB}$), or a virtual-world one ($\mathcal{V}_{RGB}$), or a virtual-to-real domain-adapted one ($\mathcal{V}_{\mathcal{G}_{\mathcal{R}},RGB}$), i.e., using a GAN-based $\mathcal{V}_{RGB} \rightarrow \mathcal{R}_{RGB}$ image translation. One view of the data (v_1) corresponds to the original RGB images of each set, while the other view (v_2) corresponds to their horizontally mirrored counterparts, indicated with the symbol " ↰ ". In the multi-modal cases, view v_1 is the same as for the single-modal case (RGB), while view v_2 corresponds to the depth (D) estimated from the RGB images by using an off-the-shelf monocular depth estimation model.

Modality	Domain Shift?	$\mathcal{X}^l_{v_1}$	$\mathcal{X}^l_{v_2}$	$\mathcal{X}^u_{v_1}$	$\mathcal{X}^u_{v_2}$
	No	$\mathcal{R}_{RGB}$	$\mathcal{R}_{RGB}{}^{↰}$		
Single-modal	Yes	$\mathcal{V}_{RGB}$	$\mathcal{V}_{RGB}{}^{↰}$	$\mathcal{R}_{RGB}$	
	Adapted	$\mathcal{V}_{\mathcal{G}_{\mathcal{R}},RGB}$	$\mathcal{V}_{\mathcal{G}_{\mathcal{R}},RGB}{}^{↰}$		
	No	$\mathcal{R}_{RGB}$	$\mathcal{R}_D$		
Multi-modal	Yes	$\mathcal{V}_{RGB}$	$\mathcal{V}_D$	$\mathcal{R}_{RGB}$	$\mathcal{R}_D$
	Adapted	$\mathcal{V}_{\mathcal{G}_{\mathcal{R}},RGB}$	$\mathcal{V}_{\mathcal{G}_{\mathcal{R}},D}$		

4. Experimental Results

4.1. Datasets and Evaluation Protocol

We follow the experimental setup of [17]. Therefore, we use KITTI [29] and Waymo [30] as real-world datasets, here denoted as $\mathcal{K}$ and $\mathcal{W}$, respectively. We use a variant of the SYNTHIA dataset [31] as virtual-world data, here denoted as $\mathcal{V}$. For $\mathcal{K}$ we use Xiang et al. [32] split, which reduces the correlation between training and testing data. While this implies that $\mathcal{K}$ is formed by isolated images, $\mathcal{W}$ is composed of image sequences. To align its acquisition conditions with $\mathcal{K}$, we consider daytime sequences without adverse weather. From them, as recommended in [30], we randomly select some sequences for training and the rest for testing. Furthermore, we adapt $\mathcal{W}$'s image size to match $\mathcal{K}$ (i.e., 1240×375 pixels) by first eliminating the top rows of each image so avoiding large sky areas, and then selecting a centered area of 1240 pixel width. The 2D BBs of $\mathcal{W}$ and $\mathcal{V}$, are obtained by projecting the available 3D BBs. On the other hand, $\mathcal{V}$ is generated by mimicking some acquisition conditions of $\mathcal{K}$, such as image resolution, non-adverse weather, daytime, and only considering isolated shots instead of image sequences. Besides, $\mathcal{V}$'s

images include standard visual post-effects such as anti-aliasing, ambient occlusion, depth of field, eye adaptation, blooming, and chromatic aberration. In the following, we term as $\mathcal{K}^{tr}$ and $\mathcal{K}^{tt}$ the training and testing sets of $\mathcal{K}$, respectively. Analogously, $\mathcal{W}^{tr}$ and $\mathcal{W}^{tt}$ are the training and testing sets of $\mathcal{W}$. For each dataset, Table 2 summarizes the number of images and object BBs (vehicles and pedestrians) used for training and testing our object detectors. Note that $\mathcal{V}$ is only used for training purposes.

Table 2. Datasets ($\mathcal{X}$): train ($\mathcal{X}^{tr}$) and test ($\mathcal{X}^{tt}$) statistics, $\mathcal{X} = \mathcal{X}^{tr} \cup \mathcal{X}^{tt}, \mathcal{X}^{tr} \cap \mathcal{X}^{tt} = \varnothing$.

Dataset ($\mathcal{X}$)	$\mathcal{X}^{tr}$			$\mathcal{X}^{tt}$		
	Images	Vehicles	Pedestrians	Images	Vehicles	Pedestrians
Virtual ($\mathcal{V}$)	19,791	43,326	44,863			
KITTI ($\mathcal{K}$)	3682	14,941	3154	3799	18,194	1333
Waymo ($\mathcal{W}$)	9873	64,446	9918	4161	24,600	3068

We apply the KITTI benchmark protocol for object detection [29]. Furthermore, following [17], we focus on the so-called moderate difficulty, which implies that the minimum BB height to detect objects is 25 pixels for $\mathcal{K}$ and 50 pixels for $\mathcal{W}$. Once co-training finishes, we use the labeled data ($\mathcal{X}^l$) and the data self-labeled by co-training ($\mathcal{X}^{\hat{l}}$) to train the final object detector, namely, ϕ_F. Since this is the ultimate goal, we use the accuracy of such a detector as metric to evaluate the effectiveness of the co-training procedure. If it performs well at self-labeling objects, the accuracy of ϕ_F should be close to the upper-bound (i.e., when the 100% of the real-world labeled data used to train ϕ_F is provided by humans), otherwise, the accuracy of ϕ_F is expected to be close to the lower-bound (i.e., when using either a small percentage of human-labeled data or only virtual-world data to train ϕ_F).

4.2. Implementation Details

When using virtual-world images we not only experiment with the originals but also with their GAN-based virtual-to-real translated counterparts, i.e., aiming at closing the domain shift between virtual and real worlds. Since the translated images are the same for both co-training modalities, we take them from [17], where a CycleGAN [19] was used to learn the translations $\mathcal{G}_{\mathcal{K}} : \mathcal{V} \to \mathcal{K}$ and $\mathcal{G}_{\mathcal{W}} : \mathcal{V} \to \mathcal{W}$. To obtain these images, CycleGAN training was done for 40 epochs using a weight of 1.0 for the identity mapping loss, and a patch-wise strategy with patches of 300×300 pixels, while keeping the rest of the parameters as recommended in [19]. We denote as $\mathcal{V}_{\mathcal{G}_{\mathcal{K}}} = \mathcal{G}_{\mathcal{K}}(\mathcal{V})$ and $\mathcal{V}_{\mathcal{G}_{\mathcal{W}}} = \mathcal{G}_{\mathcal{W}}(\mathcal{V})$ the sets of virtual-world images transformed by $\mathcal{G}_{\mathcal{K}}$ and $\mathcal{G}_{\mathcal{W}}$, respectively. The 2D BBs in $\mathcal{V}$ are used for $\mathcal{V}_{\mathcal{G}_{\mathcal{K}}}$ and $\mathcal{V}_{\mathcal{G}_{\mathcal{W}}}$. Furthermore, note that analogously to $\mathcal{V}$, $\mathcal{V}_{\mathcal{G}_{\mathcal{K}}}$ and $\mathcal{V}_{\mathcal{G}_{\mathcal{W}}}$ are only used for training. For multi-modal co-training, depth estimation is applied indistinctly to the real-world datasets, the virtual-world one, and the GAN-based translated ones.

In the multi-modal setting, one of the co-training views is the appearance (RGB) and the other is the corresponding estimated depth (D). To keep co-training single-sensor, we use monocular depth estimation (MDE). In particular, we leverage a state-of-the-art MDE model publicly released by Yin et al. [18]. It has been trained on KITTI data, thus, being ideal to work with $\mathcal{K}$. However, since our aim is not to obtain accurate depth estimation, but to generate an alternative data view useful to detect the objects of interest, we have used the same MDE model for all the considered datasets. Despite this, Figure 3 shows how the estimated depth properly captures the depth structure for the images of all datasets, i.e., not only for $\mathcal{K}$, but also for $\mathcal{W}, \mathcal{V}, \mathcal{V}_{\mathcal{G}_{\mathcal{K}}}$ and $\mathcal{V}_{\mathcal{G}_{\mathcal{W}}}$. However, we observe that the depth structure for $\mathcal{V}_{\mathcal{G}_{\mathcal{K}}}$'s and $\mathcal{V}_{\mathcal{G}_{\mathcal{W}}}$'s images is more blurred at far distances than for $\mathcal{V}$, especially for $\mathcal{V}_{\mathcal{G}_{\mathcal{W}}}$.

Figure 3. RGB images with their estimated depth. From top to bottom rows: samples from $\mathcal{K}$, $\mathcal{V}_{\mathcal{G}_{\mathcal{K}}}$, $\mathcal{V}$, $\mathcal{V}_{\mathcal{G}_{\mathcal{W}}}$, $\mathcal{W}$. The samples of $\mathcal{V}_{\mathcal{G}_{\mathcal{K}}}$ and $\mathcal{V}_{\mathcal{G}_{\mathcal{W}}}$ correspond to transforming the samples of $\mathcal{V}$ to $\mathcal{K}$ and $\mathcal{W}$ domains, respectively. The monocular depth estimation model [18] was trained on the $\mathcal{K}$ domain.

Following [17], we use Faster R-CNN with a VGG16 feature extractor (backbone) as the CNN architecture for object detection, i.e., as Φ in Algorithms 1. In particular, we rely on the Detectron implementation [33]. For training, we always initialize VGG16 with ImageNet pre-trained weights, while the weights of the rest of the CNN (i.e., the candidates' generator and classifier stages) are randomly initialized. Faster R-CNN training is based on 40,000 iterations of the SGD optimizer. Note that these iterations refer to the function $\text{Train}(\Phi, \mathcal{H}_\Phi, \mathcal{S}^l, \mathcal{S}^{\hat{l}}) : \phi$ in Algorithms 1, not to co-training cycles. Each iteration uses a mini-batch of two images randomly sampled from $\mathcal{S}^l \cup \mathcal{S}^{\hat{l}}$. Thus, looking at how $\text{Train}(\Phi, \mathcal{H}_\Phi, \mathcal{S}^l, \mathcal{S}^{\hat{l}}) : \phi$ is called in Algorithms 1, we can see that, for each view, the parameter $\mathcal{S}^l$ receives the same input in all co-training cycles, while $\mathcal{S}^{\hat{l}}$ changes from cycle-to-cycle. The SGD learning rate starts at 0.001 and we set a decay of 0.1 at iterations 30,000 and 35,000. In the case of multi-modal co-training, we use horizontal mirroring as a data augmentation technique. However, we cannot do it in the case of single-modal co-training because both data views would highly correlate. Note that, as it was done in [17] and we can see in Table 1, horizontal mirroring is the technique used to generate one of the data views in single-modal co-training. In terms of Algorithms 1, all these settings are part of $\mathcal{H}_\Phi$ and they are the same to train both ϕ_1 and ϕ_2. The values set for the co-training hyper-parameters are shown in Table 3.

Finally, note that the final detection model used for evaluations, ϕ_F, could be based on any CNN architecture for object detection, provided the GT it expects consists of 2D BBs. However, for the sake of simplicity, we also rely on Faster R-CNN to obtain ϕ_F.

Table 3. Co-training hyper-parameters as defined in Algorithms 1. We use the same values for $\mathcal{K}$ and $\mathcal{W}$ datasets, but $\mathcal{H}_{seq}$ only applies to $\mathcal{W}$. N, n, m, Δt_1, and Δt_2 are set in number-of-images units, K_{min}, K_{max} and ΔK in number-of-cycles, $T_{\Delta_{mAP}}$ runs in $[0..100]$. T hyper-parameter contains the confidence detection thresholds for vehicles and pedestrians, which run in $[0..1]$, and we have set the same value for both. The setting $m = \infty$ means that all the images self-labeled at current co-training cycle are exchanged by the models ϕ_1 and ϕ_2 for collaboration, i.e., these will then select the n less confident for them.

| | | | | | $\mathcal{H}_{stp}$ | | | $\mathcal{H}_{seq}$ | |
| | | | | | | | | | |
T	N	n	m	K_{min}	K_{max}	ΔK	$T_{\Delta_{mAP}}$	Δt_1	Δt_2
{0.8,0.8}	500	100	∞	20	30	5	2.0	5	10

4.3. Results

To include multi-modality we improved and adapted the code used in [17]. For this reason, we not only execute the multi-modal co-training experiments but also redo the single-modal and baseline ones. The conclusions in [17] remain, but by repeating these experiments, all the results presented in this paper are based on the same code.

4.3.1. Standard SSL Setting

We start the evaluation of co-training in a standard SSL setting, i.e., working only with either the $\mathcal{K}$ or $\mathcal{W}$ dataset to avoid domain shift. In this setting, the cardinality of the unlabeled dataset is supposed to be significantly higher than the cardinality of the labeled, we divide the corresponding training sets accordingly. In particular, for $\mathcal{X}^{tr} \in \{\mathcal{K}^{tr}, \mathcal{W}^{tr}\}$, we use the $p\%$ of $\mathcal{X}^{tr}$ as the labeled training set ($\mathcal{X}^l$) and the rest as the unlabeled training set ($\mathcal{X}^u$). We explore $p = 5$ and $p = 10$, where the corresponding $\mathcal{X}^{tr}$ is sampled randomly once and frozen for all the experiments. Table 4 shows the obtained results for both co-training modalities. We also report upper-bound (UB) and lower-bound (LB) results. The UB corresponds to the case $p = 100$, i.e., all the BBs are human-labeled. The LBs correspond to the $p = 5$ and $p = 10$ cases without using co-training, thus, not leveraging the unlabeled data. Although in this paper we assume that ϕ_F will be based on RGB data alone, since we use depth estimation for multi-modal co-training, as a reference we also report the UB and LB results obtained by using the estimated depth alone to train the corresponding ϕ_F.

Analyzing Table 4, we confirm that the UB and LBs based only on the estimated depth (D) show a reasonable accuracy, although not at the level of appearance (RGB) alone. This is required for the co-training to have the chance to perform well. Aside from this, we see how, indeed, both co-training modalities clearly outperform LBs. In the $p = 5$ case, multi-modal co-training clearly outperforms single-modal in all classes (V and P) and datasets ($\mathcal{K}$ and $\mathcal{W}$). Moreover, the accuracy improvement over the LBs is significantly larger than the remaining distance to the UBs. In the $p = 10$ case, both co-training modalities perform similarly. On the other hand, for $\mathcal{K}$, the accuracy of multi-modal co-training with $p = 5$ is just ~2 points below the single-modal with $p = 10$, and less than 1 point for $\mathcal{W}$. Therefore, for 2D object detection, we recommend multi-modal co-training for a standard SSL setting with a low ratio of labeled vs. unlabeled images.

Table 4. SSL (co-training) results on vehicle (V) and pedestrian (P) detection, reporting mAP. From a training set $\mathcal{X}^{tr} \in \{\mathcal{K}^{tr}, \mathcal{W}^{tr}\}$, we preserve the labeling information for a randomly chosen $p\%$ of its images, while it is ignored for the rest. We report results for $p = 100$ (all labels are used), $p = 5$ and $p = 10$. If $\mathcal{X}^{tt} = \mathcal{K}^{tt}$, then $\mathcal{X}^{tr} = \mathcal{K}^{tr}$; analogously, when $\mathcal{X}^{tt} = \mathcal{W}^{tt}$, then $\mathcal{X}^{tr} = \mathcal{W}^{tr}$, i.e., there is no domain shift in these experiments. Co-T (RGB) and Co-T (RGB/D) stand for single and multi modal co-training, respectively. UP and LB stand for upper bound and lower bound, respectively. Bold results indicate **best performing** within the block, where blocks are delimited by horizontal lines. Second best is underlined, but if the difference with the best is below 0.5 points, we use bold too. $\Delta\{\phi_{F_1} \text{ vs. } \phi_{F_2}\}$ stands for mAP of ϕ_{F_1} minus mAP of ϕ_{F_2}.

Training Set	$\mathcal{X}^{tt} = \mathcal{K}^{tt}$			$\mathcal{X}^{tt} = \mathcal{W}^{tt}$		
	V	**P**	**V&P**	**V**	**P**	**V&P**
100% Labeled (RGB)/UB	**83.43**	**67.77**	**75.60**	**61.71**	**57.74**	**59.73**
100% Labeled (D)/UB	80.80	53.43	67.12	55.14	37.67	46.41
5% Labeled (RGB)/LB	65.20	46.08	55.64	51.69	41.92	46.81
5% Labeled (D)/LB	64.45	26.70	45.58	45.21	29.98	36.70
5% Labeled + Co-T (RGB)	74.26	55.41	64.84	54.00	56.34	55.17
5% Labeled + Co-T (RGB/D)	**78.64**	**57.40**	**68.02**	**58.42**	**56.98**	**57.70**
10% Labeled (RGB)/LB	72.31	45.51	58.91	49.53	49.83	49.68
10% Labeled (D)/LB	69.54	46.31	57.93	47.93	33.98	40.96
10% Labeled + Co-T (RGB)	78.63	**60.99**	**69.81**	56.15	**60.20**	**58.18**
10% Labeled + Co-T (RGB/D)	**79.68**	60.55	70.12	59.54	57.17	58.36
$\Delta\{(5\% \text{ L.} + \text{Co-T (RGB/D)}) \text{ vs. } (5\% \text{ L. (RGB)/LB})\}$	+13.44	+11.32	+12.38	+6.73	+15.06	+10.89
$\Delta\{(5\% \text{ L.} + \text{Co-T (RGB/D)}) \text{ vs. } (100\% \text{ L. (RGB)/UB})\}$	−4.79	−10.37	−7.58	−3.29	−0.76	−2.03
$\Delta\{(10\% \text{ L.} + \text{Co-T (RGB/D)}) \text{ vs. } (10\% \text{ L. (RGB)/LB})\}$	+7.37	+15.04	+11.21	+10.01	+7.34	+8.68
$\Delta\{(10\% \text{ L.} + \text{Co-T (RGB/D)}) \text{ vs. } (100\% \text{ L. (RGB)/UB})\}$	−3.75	−7.22	−5.48	−2.17	−0.57	−1.37

4.3.2. SSL under Domain Shift

Table 5 shows the LB results for a ϕ_F fully trained on virtual-world images (source domain); the results of training only on the real-world images (target domain), where these images are 100% human-labeled (i.e., 100% Labeled RGB in Table 4); and the combination of both, which turns out to be the UB. In the case of testing on $\mathcal{W}^{tt}$ and having $\mathcal{V}$ involved in the training, we need to accommodate the different labeling style (mainly the margin between BBs and objects) of $\mathcal{W}^{tt}$ and $\mathcal{V}$. This is only needed for a fair quantitative evaluation, thus, for performing such evaluation the detected BBs are resized by per-class constant factors. However, the qualitative results presented in the rest of the paper are shown directly as they come by applying the corresponding ϕ_F, i.e., without applying any resizing. On the other hand, this resizing is not needed for $\mathcal{K}^{tt}$ since its labeling style is similar enough to $\mathcal{V}$.

Table 5. SSL (co-training) results on vehicle (V) and pedestrian (P) detection, under domain shift, reported as mAP. $\mathcal{X}^l$ refers to the human-labeled target-domain training set; thus, if $\mathcal{X}^{tt} = \mathcal{K}^{tt}$, then $\mathcal{X}^l = \mathcal{K}^{tr}$, and if $\mathcal{X}^{tt} = \mathcal{W}^{tt}$, then $\mathcal{X}^l = \mathcal{W}^{tr}$. $\mathcal{X}^{\hat{l}}$ consists of the same images as $\mathcal{X}^l$, but self-labeled by co-training. Co-T (RGB), Co-T (RGB/D), UP, LB, $\Delta\{\phi_{F_1} \text{ vs. } \phi_{F_2}\}$, bold and underlined numbers are analogous to those in Table 4.

Training Set	$\mathcal{X}^{tt} = \mathcal{K}^{tt}$			$\mathcal{X}^{tt} = \mathcal{W}^{tt}$		
	V	**P**	**V&P**	**V**	**P**	**V&P**
Source ($\mathcal{V}$)/LB	67.46	65.18	66.32	38.88	53.37	46.13
Target ($\mathcal{X}^l$)	83.43	67.77	75.60	**61.71**	57.74	59.73
Target + Source ($\mathcal{X}^l \& \mathcal{V}$)/UB	**87.15**	**74.69**	**80.92**	59.97	**62.86**	**61.42**
Co-T (RGB) + Source ($\mathcal{X}^{\hat{l}} \& \mathcal{V}$)	77.97	**71.32**	74.65	48.56	56.33	52.45
Co-T (RGB/D) + Source ($\mathcal{X}^{\hat{l}} \& \mathcal{V}$)	**82.90**	67.36	**75.13**	**64.40**	**59.17**	**61.79**
$\Delta\{(\text{Co-T (RGB/D)} + \text{Source}) \text{ vs. LB}\}$	+15.44	+2.18	+8.81	+25.52	+5.80	+15.66
$\Delta\{(\text{Co-T (RGB/D)} + \text{ASource}) \text{ vs. UB}\}$	−4.25	−7.33	−5.79	−0.16	−4.27	−2.21

According to Table 5, both co-training modalities significantly outperform the LB. Again, multi-modal co-training outperforms single-modal, especially on vehicles. Comparing multi-modal co-training with the LB, we see improvements of ∼15 points for vehicles in $\mathcal{K}$, and ∼25 in $\mathcal{W}$. Considering the joint improvement for vehicles and pedestrians we see ∼8 points for $\mathcal{K}$, and ∼15 for $\mathcal{W}$, while the distances to the UB are of ∼5 points for $\mathcal{K}$, and ∼2 for $\mathcal{W}$. Therefore, for 2D object detection, we recommend multi-modal co-training for an SSL scenario where the labeled data comes from a virtual world, i.e., when no human labeling is required at all, but there is a virtual-to-real domain shift.

4.3.3. SSL after GAN-Based Virtual-to-Real Image Translation

Table 6 is analogous to Table 5, just changing the original virtual-world images ($\mathcal{V}$) by their GAN-based virtual-to-real translated counterparts ($\mathcal{V}_{\mathcal{G}_{\mathcal{K}}}/\mathcal{V}_{\mathcal{G}_{\mathcal{W}}}$). In the case of testing on $\mathcal{W}^{tt}$ and having $\mathcal{V}_{\mathcal{G}_{\mathcal{W}}}$ involved in the training, we apply the BB resizing mentioned in Section 4.3.2 for the quantitative evaluation. Focusing on the V&P results, we see that both the UB and LB of Table 6 show higher accuracy than in Table 5, which is due to the reduction of the virtual-to-real domain shift achieved thanks to the use of $\mathcal{V}_{\mathcal{G}_{\mathcal{K}}}/\mathcal{V}_{\mathcal{G}_{\mathcal{W}}}$. Still, co-training enables to improve the accuracy of the LBs, almost reaching the accuracy of the UBs. For instance, in the combined V&P detection accuracy, the single-modal co-training is 1.66 points behind the UB for $\mathcal{K}$, and 3.59 for $\mathcal{W}$. Multi-modal co-training is 2.63 points behind the UB for $\mathcal{K}$, and 4.01 for $\mathcal{W}$. Thus, in this case, single-modal co-training is performing better than multi-modal. Therefore, for 2D object detection, we can recommend even single-modal co-training for an SSL scenario where the labeled data comes from a virtual world but a properly trained GAN can perform virtual-to-real domain adaptation. On the other hand, in the case of $\mathcal{W}$, co-training from $\mathcal{V}_{\mathcal{G}_{\mathcal{W}}}$ gives rise to worse results than by using $\mathcal{V}$. We think this is due to a worse depth estimation (see Figure 3). In general, this suggests that whenever it is possible, training a specific monocular depth estimator for the unlabeled real-world data may be beneficial for multi-modal co-training (recent advances on vision-based self-supervision for monocular depth estimation [10,34] can be a good starting point). For this particular case, training the virtual-to-real domain adaptation GAN simultaneously to the monocular depth estimation CNN could be an interesting idea to explore in the future (we can leverage inspiration from [35,36]).

Table 6. SSL (co-training) results on vehicle (V) and pedestrian (P) detection, after GAN-based virtual-to-real image translation, reported as mAP. ASource (adapted source) refers to $\mathcal{V}_{\mathcal{G}} \in \{\mathcal{V}_{\mathcal{G}_{\mathcal{K}}}, \mathcal{V}_{\mathcal{G}_{\mathcal{W}}}\}$. $\mathcal{X}^l$, $\mathcal{X}^{\hat{l}}$, Source, Co-T (RGB), Co-T (RGB/D), UP, LB, $\Delta\{\phi_{F_1}$ vs. $\phi_{F_2}\}$, bold and underlined numbers are analogous to those in Table 5.

Training Set	$\mathcal{X}^{tt} = \mathcal{K}^{tt}$			$\mathcal{X}^{tt} = \mathcal{W}^{tt}$		
	V	**P**	**V&P**	**V**	**P**	**V&P**
ASource ($\mathcal{V}_{\mathcal{G}}$)/LB	78.41	65.39	71.90	52.60	56.36	54.48
Target ($\mathcal{X}^l$)	<u>83.43</u>	<u>67.77</u>	<u>75.60</u>	<u>61.71</u>	<u>57.74</u>	<u>59.73</u>
Target + ASource ($\mathcal{X}^l$&$\mathcal{V}_{\mathcal{G}}$)/UB	**86.82**	**71.59**	**79.21**	**64.56**	**63.44**	**64.00**
Co-T (RGB) + ASource ($\mathcal{X}^{\hat{l}}$&$\mathcal{V}_{\mathcal{G}}$)	**85.17**	**69.93**	**77.55**	**61.49**	**59.33**	**60.41**
Co-T (RGB/D) + ASource ($\mathcal{X}^{\hat{l}}$&$\mathcal{V}_{\mathcal{G}}$)	83.68	**69.48**	76.58	**61.49**	58.49	**59.99**
$\Delta\{$(Co-T (RGB) + ASource) vs. LB$\}$	+6.76	+4.54	+5.65	+8.89	+2.97	+5.93
$\Delta\{$(Co-T (RGB/D) + ASource) vs. LB$\}$	+5.27	+4.09	+4.68	+8.89	+2.13	+5.51
$\Delta\{$(Co-T (RGB) + ASource) vs. UB$\}$	−1.65	−1.66	−1.66	−3.07	−4.11	−3.59
$\Delta\{$(Co-T (RGB/D) + ASource) vs. UB$\}$	−3.14	−2.11	−2.63	−3.07	-4.95	−4.01

4.3.4. Analyzing Co-Training Cycles

Figures 4 and 5 illustrate how co-training strategies would perform as a function of the stopping cycle, for a standard SSL setting (Figure 4), as well as under domain shift (Source) and when this is reduced (ASource) by using $\mathcal{V}_{\mathcal{G}_{\mathcal{K}}}/\mathcal{V}_{\mathcal{G}_{\mathcal{W}}}$ (Figure 5). We take the

self-labeled images at different co-training cycles (*x*-axis) as if these cycles were determined to be the stopping ones. The labeled images together with the self-labeled by co-training up to the indicated cycle are used to train the corresponding ϕ_F. Then, we plot (*y*-axis) the accuracy (mAP) of each ϕ_F in the corresponding testing set, i.e., either $\mathcal{K}^{tt}$ or $\mathcal{W}^{tt}$. We can see how co-training strategies allow improving over the LBs from early iterations and, although slightly oscillating, keep improving until stabilization is reached. No drifting to erroneous self-labeling is observed. At this point, the object samples which remain as unlabeled but are required to reach the maximum accuracy, probably are too different in some aspect from the labeled and self-labeled ones (e.g., they may be under a too-heavy occlusion) and would never be self-labeled without additional information. Then, combining co-training with active learning (AL) cycles could be an interesting alternative, since occasional human loops could help co-training to progress more. We see also how when the starting point for co-training is at a lower accuracy, multi-modal co-training usually outperforms single-modal (e.g., in the 5% setting and under domain shift).
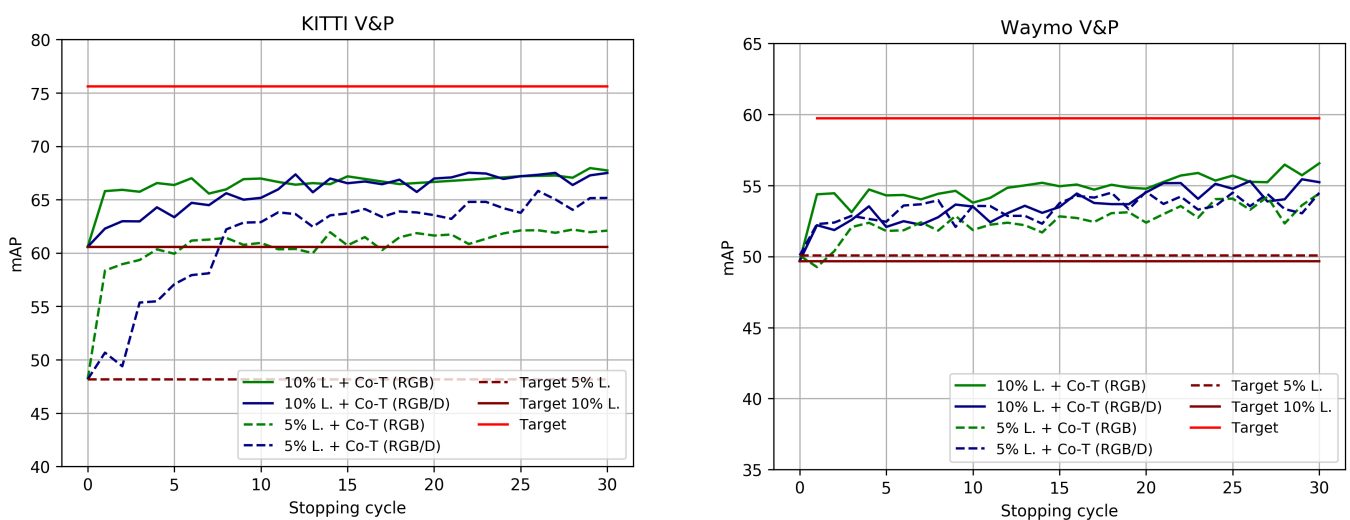

Figure 4. V&P detection accuracy of co-training approaches as a function of the stopping cycle. Co-T (RGB) and Co-T (RGB/D) refer to single and multi modal co-training, respectively. Target refers to the use of the 100% labeled training data, while Target *p*% L. indicates a lower percentage $p \in \{5, 10\}$ of labeled data available for training. Accordingly, *p*% L. + Co-T (view), view ∈{RGB, RGB/D}, are combinations of those. These plots complement the results shown in Table 4.
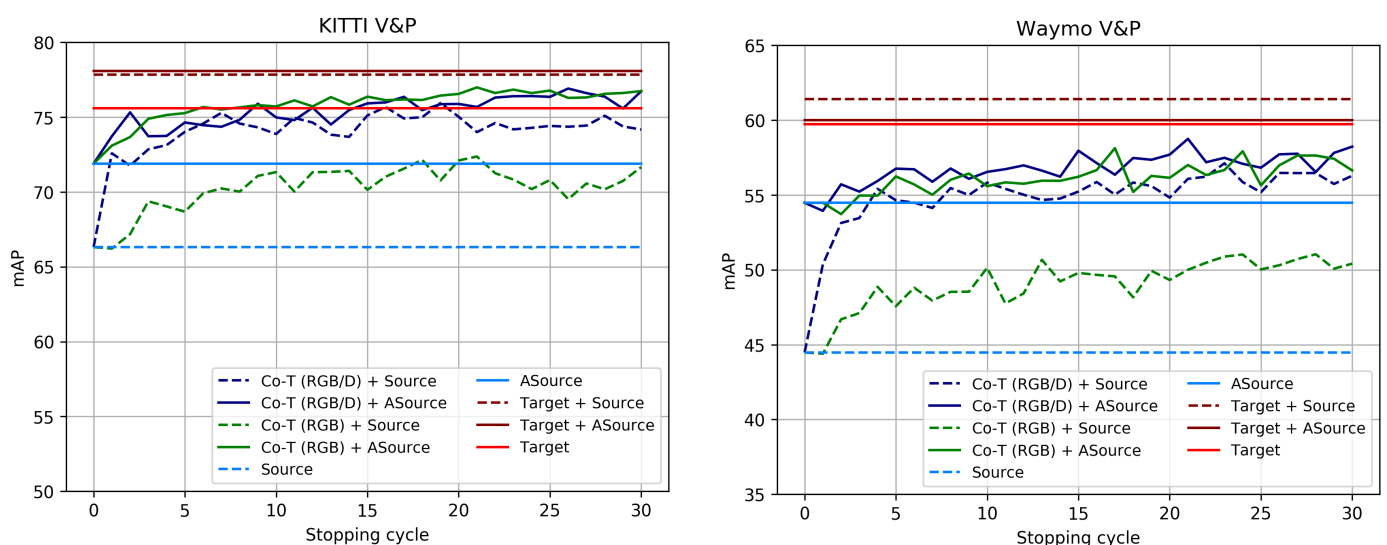

Figure 5. V&P detection accuracy of co-training approaches as a function of the stopping cycle. These plots are analogous to those in Figure 4 for the cases of using virtual-world data, i.e., both with domain shift (Source) and reducing it by the use of GANs (ASource). The Targets are the same as in Figure 4. These plots complement the results shown in Tables 5 and 6.

4.3.5. Qualitative Results

Figures 6 and 7 present qualitative results for ϕ_F's trained after stopping co-training at cycles 1, 10, 20 and when it stops automatically (i.e., the stopping condition of the loop in Algorithms 1 becomes true). The shown examples correspond to the most accurate setting for each dataset; i.e., for $\mathcal{K}$ (Figure 6) this is the co-training from $\mathcal{V}_{\mathcal{G}_\mathcal{K}}$ no matter the modality, while for $\mathcal{W}$ (Figure 7) this is the co-training from $\mathcal{V}$ in the multi-modal case and from $\mathcal{V}_{\mathcal{G}_\mathcal{W}}$ in the single-modal. Note that Tables 4–6, suggest to combine co-training with virtual-world data to obtain more accurate ϕ_F's.

Figure 6. Qualitative results of how ϕ_F would perform on $\mathcal{K}^{tt}$ by stopping co-training at different cycles. We focus on co-training and object detection working from $\mathcal{V}_{\mathcal{G}_\mathcal{K}}$ (ASource). There are three blocks of results vertically arranged. At each block, the top-left image shows the results when using the 100% human-labeled training data plus $\mathcal{V}_{\mathcal{G}_\mathcal{K}}$ (Target + ASource), i.e., UB results. Detection results are shown as green BBs, and GT as red BBs. The top-right image of each block shows the results that we would obtain without leveraging the unlabeled data (ASource), i.e., LB results. The rest of the rows of the block, from top-second to bottom, correspond to stopping co-training at cycles 1, 10, 20, and automatically. In these rows, the images at the left column correspond to multi-modal co-training (i.e., Co-T (RGB/D)) and those at the right column to single-modal co-training (i.e., Co-T (RGB)).

In the left block of Figure 6, we show a case where both co-training modalities perform similarly on pedestrian detection, with final detections (green BBs) very close to the GT (red BBs), and clearly better than if we do not leverage the unlabeled data (top-right image of the block). We see also that the results are very similar to the case of using the 100% of human-labeled data (top-left image of the block). Moreover, even from the initial cycles of both co-training modalities the results are reasonably good, although, the best is expected when co-training finishes automatically (bottom row of the block), i.e., after the minimum number of cycles is exceeded ($K_{min} = 20$ in Table 3). In the mid-block, we see that only multi-modal co-training helps to properly detect a very close and partially occluded vehicle.

In the right block, only multi-modal co-training helps to keep and improve the detection of a close pedestrian. Both co-training modalities help to keep an initially detected van, but multi-modal co-training induces a better BB adjustment. This is an interesting case. Since $\mathcal{V}$ only contains different types of cars but lacks a meaningful number of van samples, and $\mathcal{K}$ only has a very small percentage of those labeled, we have focused our study on the different types of cars. Therefore, vans are neither considered for training nor testing, i.e., their detection or misdetection does not affect the mAP metric either positively or negatively. However, co-training is an automatic self-labeling procedure, thus it may capture or keep these samples and then force training with them. Moreover, in this setting, the hard-negatives are mined only from the virtual-world images (translated or not by a GAN) since they are fully labeled. Thus, if no sufficient vans are part of the virtual-world images, these objects cannot act as hard negatives, so that they may be detected or misdetected depending on their resemblance to the targeted objects (here types of cars). We think this is the case here. Thus, this is an interesting consideration for designing future co-training procedures supported by virtual-world data. Alternatively, by complementing co-training with occasional AL cycles, these special false positives could be reported by the human in the AL loop (provided we really want to treat them as false positives). On the other hand, in the same block of results, we see also a misdetection (isolated red BB), which does account for the quantitative evaluation. It corresponds to a rather occluded vehicle which is not detected even when relying on human labeling (top-left image of the block). Finally, note the large range of detection distances achieved for vehicles.

In the left block of Figure 7, we see even a larger detection range for the detected vehicles than in Figure 6. Faraway vehicles (small green BBs) are considered as false positives for the qualitative evaluation because these are not part of the $\mathcal{W}^{tt}$ GT (since they do not have labeled 3D BBs from which the 2D BBs are obtained). Thanks to the use of virtual-world data, these vehicles are detected (second row of the block) and both co-training modalities do not damage their detection. Note how the UBs based on virtual-world data and human-labeled real-world data are not able to detect such vehicles (first row of the block) because human labeling did not consider these faraway vehicles, while co-training does consider them as such. Besides, multi-modal co-training enables the detection of the closer vehicle since cycle 10. In the next block to the right, multi-modal co-training enables to detect a close kid since cycle 10, while single-modal does not at the end. In addition, single-modal co-training also introduces a distant false positive. Similarly to the left block, in this block both co-training modalities keep an unlabeled vehicle detected thanks to the use of the virtual-world data (second row), not detected (first row) when these data are complemented with human-labeled data (since, again, this vehicle is not even labeled). What is happening in these cases, is that there is a lack of real-world human-labeled 3D BBs for distant vehicles, which is compensated by the use of virtual-world data and maintained by co-training. In the next block to the right, we see how a pedestrian is detected thanks to both co-training methods since only using virtual-world data was not possible (second row). In the right block, both co-training modalities allow for vehicle and pedestrian detections similar to the UBs (first row). Note that the vehicle partially hidden behind the pedestrian was not detected by only using virtual-world data (second row), and neither was detected the pedestrian when using $\mathcal{V}$ (second row, left) or was poorly detected when using $\mathcal{V}_{\mathcal{G}_{\mathcal{W}}}$ (second row, right).

Finally, Figure 8 shows additional qualitative results on $\mathcal{K}^{tt}$ and $\mathcal{W}^{tt}$ when using multi-modal co-training, in the case of $\mathcal{K}^{tt}$ based on $\mathcal{V}_{\mathcal{G}_{\mathcal{K}}}$ and $\mathcal{V}$ for $\mathcal{W}^{tr}$, i.e., we show the results of the respective best models. Overall, in the case of $\mathcal{K}^{tt}$, we see how multi-modal co-training (Co-T (RGB/D)) enables to better adjust detection BBs, and removing some false positives. In the case of $\mathcal{W}^{tt}$, multi-modal co-training enables to keep even small vehicles that are not part of the GT but are initially detected thanks to the use of virtual-world data. It also helps to detect vehicles and pedestrians not detected by only using the virtual-world data, although further improvements are needed since some pedestrians are still difficult to detect even with co-training.

Figure 7. Qualitative results similar to those in Figure 6, but testing on $\mathcal{W}^{tt}$, co-training from $\mathcal{V}$ in the multi-modal case (left column of each block), and $\mathcal{V}_{\mathcal{G}_\mathcal{W}}$ in the single-modal case (right column of each block). Since, in these examples, the two co-training modalities are based on different (labeled) data, the first row of each block shows the respective UB results, i.e., those based on training with $\mathcal{W}^{tr}$ and either with $\mathcal{V}$ (left image: Target + Source) or $\mathcal{V}_{\mathcal{G}_\mathcal{W}}$ (right image: Target + ASource). The second row of each block shows the respective results we would obtain without leveraging the unlabeled data, i.e., the LBs based on training with $\mathcal{V}$ (left image: Source) or $\mathcal{V}_{\mathcal{G}_\mathcal{W}}$ (right image: ASource). As in Figure 6, the rest of the rows of each block correspond to stopping co-training at cycles 1, 10, 20, and automatically.

4.3.6. Answering (Q1) and (Q2)

After presenting our multi-modal co-training and the extensive set of experiments carried out, we can answer the research questions driving this study. In particular, we base our answers in the quantitative results presented in Tables 4–6, the plots shown in Figures 4 and 5, as well as the qualitative examples shown in Figures 6–8, together with the associated comments we have drawn from them.

(Q1) Is multi-modal (RGB/D) co-training effective on the task of providing pseudo-labeled object BBs? Indeed, multi-modal co-training is effective for self-labeling object BBs under different settings, namely, for standard SSL (no domain shift, a few human-labeled data) and when using virtual-world data (many virtual-world labeled data, but no human-labeled data) both under domain shift and after reducing it by GAN-based virtual-to-real image translation. The achieved improvement over the lower bound configurations is significant, allowing to be almost in pair with upper bound configurations. In the standard SSL setting, by only labeling the 5% of the training dataset, multi-modal co-training allows obtaining accuracy values relatively close to the upper bounds. When using virtual-world data, i.e., without human labeling at all, the same observations hold. Moreover, multi-

modal co-training and GAN-based virtual-to-real image translation have been shown to complement each other.

Figure 8. Qualitative results on $\mathcal{K}^{tt}$ (top block of rows) and $\mathcal{W}^{tt}$ (bottom block of rows). In each block, we show (top row) GT as red BBs, (mid row) detections, as green BBs, when training with $\mathcal{X}^l$, (bottom row) detections with $\mathcal{X}^l \cup \mathcal{X}^{\hat{l}}$. In this case, $\mathcal{X}^{\hat{l}}$ comes from applying C-T (RGB/D) on either $\mathcal{K}^{tr}$ or $\mathcal{W}^{tr}$, and $\mathcal{X}^l$ is $\mathcal{V}_{\mathcal{G}_\mathcal{K}}$ for $\mathcal{K}^{tr}$, while it is $\mathcal{V}$ for $\mathcal{W}^{tr}$.

(Q2) How does perform multi-modal (RGB/D) co-training compared to single-modal (RGB)? We conclude that in a standard SSL setting (no domain shift, a few human-labeled data) and under virtual-to-real domain shift (many virtual-world labeled data, no human-labeled data) multi-modal co-training outperforms single-modal. In the latter case, when GAN-based virtual-to-real image translation is performed both co-training modalities are on par; at least, by using an off-the-shelf monocular depth estimation model not specifically trained on the translated images.

To drive future research, we have performed additional experiments. These consist in correcting the pseudo-labels obtained by multi-modal co-training in three different ways, namely, removing false positives (FP), adjusting the BBs to the ones of the GT (BB) for correctly self-labeled objects (true positives), and a combination of both (FP + BB). After changing the pseudo-labels in that way, we train the corresponding ϕ_F models and evaluate them. Table 7 presents the quantitative results. Focusing on the standard SSL setting (5%, 10%), we see that the main problem for vehicles in $\mathcal{K}$ is BB adjustment, while for pedestrians is the introduction of FPs. In the latter case, false negatives (FN; i.e., missing self-labeled objects) seem to be also an issue to reach upper bound accuracy. When we have the support of virtual-world data, FNs do not seem to be a problem, and addressing BB

correction for vehicles and removing FPs for pedestrians would allow reaching upper bounds. In the case of $\mathcal{W}$, we came to the same conclusions for vehicles, the main problem is BB adjustment, while in the case of pedestrians the main problem is not that clear. In other words, there is more balance between FP and BB. On the other hand, regarding these additional experiments, we trust more the conclusions derived from $\mathcal{K}$. The reason is that, as we have seen in Figures 7 and 8, co-training was correctly self-labeling objects that are not part of the GT, so in this study, these are either considered FPs and so wrongly removed (FP, FP + BB settings), or would not have a GT BB to which adjust them (BB, FP + BB settings).

Table 7. Digging in the results throw three post-processing settings applied to co-training pseudo-labels: (FP) where we remove the false positive pseudo-labels; (BB) where we change the pseudo-labels by the corresponding GT (i.e., in terms of Figures 6–8, green BBs are replaced by red ones); (FP + BB) which combines both. This table follows the terminology of Tables 4–6. Δ_X, $X \in$ {FP, BB, FP + BB}, stands for difference of setting X minus the respective original (i.e., using the co-training pseudo-labels). Moreover, for each block of results, we add the #FP/FP% row, where #FP refers to the total number of false positives that are used to train the final object detector, ϕ_F, while FP% indicates what percentage they represent regarding the whole set (labeled and self-labeled BBs) used to train ϕ_F.

Training Set	$\mathcal{X}^{tt} = \mathcal{K}^{tt}$			$\mathcal{X}^{tt} = \mathcal{W}^{tt}$		
	V	**P**	**V&P**	**V**	**P**	**V&P**
Target + ASource (UB)	86.82	71.59	79.21	64.56	63.44	64.00
5% Labeled + Co-T (RGB/D)	78.64	57.40	68.02	58.42	56.98	57.70
5% Labeled + Co-T (RGB/D)/FP	79.29	**60.50**	69.90	59.28	55.89	57.59
5% Labeled + Co-T (RGB/D)/BB	**85.18**	58.87	**72.03**	**63.25**	56.58	**59.92**
5% Labeled + Co-T (RGB/D)/FP + BB	**85.61**	58.75	**72.18**	62.49	**57.63**	**60.06**
Δ_{FP}	+0.65	+3.10	+1.88	+0.86	−1.09	−0.11
Δ_{BB}	+6.54	+1.47	+4.01	+4.83	−0.40	+2.22
Δ_{FP+BB}	+6.97	+1.35	+4.16	+4.07	+0.91	+2.36
#FP/FP%	1723/13.65%	275/16.33%		5952/10.39%	731/11.33%	
10% Labeled + Co-T (RGB/D)	79.68	60.55	70.12	59.54	**57.17**	58.36
10% Labeled + Co-T (RGB/D)/FP	79.81	**61.65**	70.73	60.24	**57.38**	58.81
10% Labeled + Co-T (RGB/D)/BB	**85.28**	59.21	**72.25**	**63.01**	56.07	59.54
10% Labeled + Co-T (RGB/D)/FP + BB	83.23	61.03	**72.13**	63.20	**56.99**	**60.10**
Δ_{FP}	+0.13	+1.10	+0.61	+0.70	+0.21	+0.45
Δ_{BB}	+5.60	−1.34	+2.01	+3.47	−1.1	+1.18
Δ_{FP+BB}	+3.55	+0.48	+2.01	+3.66	−0.18	+1.74
#FP/FP%	1998/14.06%	408/16.83%		4553/7.42%	547/7.30%	
Co-T (RGB/D) + Source	82.90	67.36	75.13	**64.40**	**59.17**	**61.79**
Co-T (RGB/D) + Source/FP	83.37	70.95	77.16	57.68	56.04	56.86
Co-T (RGB/D) + Source/BB	**88.94**	61.69	75.32	62.14	57.22	59.68
Co-T (RGB/D) + Source/FP + BB	**89.07**	**71.88**	80.48	62.56	56.59	59.58
Δ_{FP}	+0.47	+3.59	+2.03	−6.72	−3.13	−4.93
Δ_{BB}	+6.04	−5.67	+0.19	−2.26	−1.95	−2.11
Δ_{FP+BB}	+6.17	+4.52	+5.35	−1.84	−2.58	−2.21
#FP/FP%	3281/3.57%	883/0.87%		18293/21.06%	970/2.02%	
Co-T (RGB/D) + ASource	83.68	69.48	76.58	61.49	**59.33**	**60.41**
Co-T (RGB/D) + ASource/FP	83.07	70.40	76.74	60.67	57.72	59.20
Co-T (RGB/D) + ASource/BB	**89.27**	68.63	78.95	62.06	57.64	59.85
Co-T (RGB/D) + ASource/FP + BB	88.93	**71.45**	80.19	**64.67**	55.27	**59.97**
Δ_{FP}	−0.61	+0.92	0.16	−0.82	−1.61	−1.21
Δ_{BB}	+5.59	−0.85	+3.61	+0.57	−1.69	−0.56
Δ_{FP+BB}	+5.25	+1.97	+3.61	+3.18	−4.06	−0.44
#FP/FP%	3097/5.59%	479/1.03%		20949/23.15%	816/1.70%	

After this analysis, we think we can explore two main future lines of research. First, to improve BB adjustment, we could complement multi-modal co-training with instance segmentation, where using Mask R-CNN [37] would be a natural choice. Note that virtual-world data can also have instance segmentation as part of their GT suite. Second, to remove FPs, we could add an AL loop where humans could remove even several FP with a few clicks (note that this is much easier than delineating object BBs). On the other hand, additional CNN models could be explored to avoid FPs as a post-processing step to multi-modal co-training. Besides these ideas, we think that, whenever is possible, the monocular depth estimation model should be trained on the target domain data, rather than trying to use an off-the-shelf model. Since we think that not doing so was damaging the combination of multi-modal co-training and GAN-based virtual-to-real image translation, an interesting approach would be to perform both tasks simultaneously.

5. Conclusions

In this paper, we have addressed the curse of data labeling for onboard deep object detection. In particular, following the SSL paradigm, we have proposed multi-modal co-training for object detection. This co-training relies on a data view based on appearance (RGB) and another based on estimated depth (D), the latter obtained by applying monocular depth estimation, so keeping co-training as a single-sensor method. We have performed an exhaustive set of experiments covering the standard SSL setting (no domain shift, a few human-labeled data) as well as the settings based on virtual-world data (many virtual-world labeled data, no human-labeled data) both with domain shift and without (using GAN-based virtual-to-real image translation). In these settings, we have compared multi-modal co-training and appearance-based single-modal co-training. We have shown that multi-modal co-training is effective in all settings. In the standard SSL setting, from a 5% of human-labeled training data, co-training can already lead to a final object detection accuracy relatively close to upper bounds (i.e., with the 100% of human labeling). The same observation holds when using virtual-world data, i.e., without human labeling at all. Multi-modal co-training outperforms single-modal in standard SSL and under domain shift, while both co-training modalities are on par when GAN-based virtual-to-real image translation is performed; at least, by using an off-the-shelf depth estimation model not specifically trained on the translated images. Moreover, multi-modal co-training and GAN-based virtual-to-real image translation have been proved to be complementary. For the future, we plan several lines of work, namely, improving the adjustment of object BBs by using instance segmentation upon detection and removing false-positive pseudo-labels by using a post-processing AL cycle. Moreover, we believe that the monocular depth estimation model should be trained based on target domain data whenever possible. When GAN-based image translation is required, we could jointly train the monocular depth estimation model and the GAN on the target domain. Besides, we would like to extend co-training experiments to other classes of interest for onboard perception (traffic signs, motorbikes, bikes, etc.), as well as adapting the method to tackle other tasks such as pixel-wise semantic segmentation.

References

1. Sharma, N.; Jain, V.; Mishra, A. An analysis Of convolutional neural networks for image classification. *Procedia Comput. Sci.* **2018**, *132*, 377–384. [CrossRef]
2. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. In Proceedings of the Neural Information Processing Systems (NeurIPS), Montreal, QC, Canada, 7–12 December 2015.
3. Tang, Y.; Chen, M.; Wang, C.; Luo, L.; Li, J.; Lian, G.; Zou, X. Recognition and Localization Methods for Vision-Based Fruit Picking Robots: A Review. *Front. Plant Sci.* **2020**, *11*, 510. [CrossRef] [PubMed]
4. Xie, E.; Sun, P.; Song, X.; Wang, W.; Liang, D.; Shen, C.; Luo, P. PolarMask: Single shot instance segmentation with polar representation. In Proceedings of the the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020.
5. Wang, J.; Sun, K.; Cheng, T.; Jiang, B.; Deng, C.; Zhao, Y.; Liu, D.; Mu, Y.; Tan, M.; Wang, X.; et al. Deep high-resolution representation learning for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**. [CrossRef] [PubMed]
6. De Queiroz Mendes, R.; Ribeiro, E.G.; dos Santos Rosa, N.; Grassi, V., Jr. On deep learning techniques to boost monocular depth estimation for autonomous navigation. *Robot. Auton. Syst.* **2021**, *136*, 103701. [CrossRef]
7. Kang, Z.; Yang, J.; Yang, Z.; Cheng, S. A Review of Techniques for 3D Reconstruction of Indoor Environments. *Int. J. Geo-Inf.* **2020**, *9*, 330. [CrossRef]
8. Sun, C.; Shrivastava, A.; Singh, S.; Gupta, A. Revisiting unreasonable effectiveness of data in deep learning Era. In Proceedings of the International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017.
9. Kolesnikov, A.; Zhai, X.; Beyer, L. Revisiting self-supervised visual representation learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019.
10. Godard, C.; Mac Aodha, O.; Firman, M.; Brostow, G.J. Digging into self-supervised monocular depth estimation. In Proceedings of the International Conference on Computer Vision (ICCV), Seoul, Korea, 27 October–2 November 2019.
11. Settles, B. *Active Learning*; Synthesis Lectures on Artificial Intelligence and Machine Learning; Morgan & Claypool Publishers: San Rafael, CA, USA, 2012.
12. Roy, S.; Unmesh, A.; Namboodiri, V. Deep active learning for object detection. In Proceedings of the British Machine Vision Conference (BMVC), Newcastle, UK, 3–6 September 2018.
13. Chapelle, O.; Schölkopf, B.; Zien, A. *Semi-Supervised Learning*; The MIT Press: Cambridge, MA, USA, 2006.
14. Van Engelen, J.; Hoos, H. A survey on semi-supervised learning. *Mach. Learn.* **2020**, *109*, 373–440.
15. Blum, A.; Mitchell, T. Combining labeled and unlabeled data with co-Training. In Proceedings of the Conference on Computational Learning Theory (COLT), Madison, WI, USA, 24–26 July 1998.
16. Guz, U.; Hakkani-Tür, D.; Cuendet, S.; Tur, G. Co-training using prosodic and lexical information for sentence segmentation. In Proceedings of the Conference of the International Speech Communication Association (INTERSPEECH), Antwerp, Belgium, 27–31 August 2007.
17. Villalonga, G.; López, A. Co-training for on-board deep object detection. *IEEE Accesss* **2020**, *8*, 194441–194456. [CrossRef]
18. Yin, W.; Liu, Y.; Shen, C.; Yan, Y. Enforcing geometric constraints of virtual normal for depth prediction. In Proceedings of the International Conference on Computer Vision (ICCV), Seoul, Korea, 27 October–2 November 2019.
19. Zhu, J.; Park, T.; Isola, P.; Efros, A. Unpaired image-to-image translation using cycle-consistent adversarial networks. In Proceedings of the International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017.
20. Yarowsky, D. Unsupervised word sense disambiguation rivaling supervised methods. In Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL), Cambridge, MA, USA, 26–30 June 1995.
21. Jeong, J.; Lee, S.; Kim, J.; Kwak, N. Consistency-based semi-supervised learning for object detection. In Proceedings of the Neural Information Processing Systems (NeurIPS), Vancouver, BC, Canada, 8–14 December 2019.
22. Lokhande, V.S.; Tasneeyapant, S.; Venkatesh, A.; Ravi, S.N.; Singh, V. Generating accurate pseudo-labels in semi-supervised learning and avoiding overconfident predictions via Hermite polynomial activations. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020.
23. Tur, G. Co-adaptation: Adaptive co-training for semi-supervised learning. In Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Taipei, Taiwan, 19–24 April 2009.
24. Zhou, Z.; Li, M. Semi-supervised learning by disagreement. *Knowl. Inf. Syst.* **2010**, *24*, 415–439.
25. Qiao, S.; Shen, W.; Zhang, Z.; Wang, B.; Yuille, A. Deep co-Training for semi-supervised image recognition. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018.
26. Díaz, G.; Peralta, B.; Caro, L.; Nicolis, O. Co-Training for Visual Object Recognition Based on Self-Supervised Models Using a Cross-Entropy Regularization. *Entropy* **2021**, *23*, 423. [CrossRef] [PubMed]

27. Han, B.; Yao, Q.; Yu, X.; Niu, G.; Xu, M.; Hu, W.; Tsang, I.W.; Sugiyama, M. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In Proceedings of the Neural Information Processing Systems (NeurIPS), Montreal , QC, Canada, 3–8 December 2018.

28. Yu, X.; Han, B.; Yao, J.; Niu, G.; Tsang, I.; Sugiyama, M. How does disagreement help generalization against label corruption? In Proceedings of the International Conference on Machine Learning (ICML), Long Beach, CA, USA, 10–15 June 2019.

29. Geiger, A.; Lenz, P.; Urtasun, R. Are we ready for autonomous driving? The KITTI vision benchmark suite. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012.

30. Sun, P.; Kretzschmar, H.; Dotiwalla, X.; Chouard, A.; Patnaik, V.; Tsui, P.; Guo, J.; Zhou, Y.; Chai, Y.; Caine, B.; et al. Scalability in perception for autonomous driving: Waymo open dataset. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 14–19 June 2020.

31. Ros, G.; Sellart, L.; Materzyska, J.; Vázquez, D.; López, A. The SYNTHIA Dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016.

32. Xiang, Y.; Choi, W.; Lin, Y.; Savarese, S. Data-driven 3D voxel patterns for object category recognition. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015.

33. Girshick, R.; Radosavovic, I.; Gkioxari, G.; Dollár, P.; He, K. Detectron. 2018. Available online: https://github.com/facebookresearch/detectron (accessed on 20 October 2020).

34. Gurram, A.; Tuna, A.F.; Shen, F.; Urfalioglu, O.; López, A.M. Monocular depth estimation through virtual-world supervision and real-world SfM self-Supervision. *arXiv* **2021**, arXiv:2103.12209.

35. Zhao, S.; Fu, H.; Gong, M.; Tao, D. Geometry-aware symmetric domain adaptation for monocular depth estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019.

36. Pnvr, K.; Zhou, H.; Jacobs, D. SharinGAN: Combining Synthetic and Real Data for Unsupervised Geometry Estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 14–19 June 2020.

37. He, K.; Gkioxari, G.; Dollar, P.; Girshick, R. Mask R-CNN. In Proceedings of the International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017.